GUARANTEE !

『*4 Hours*』

Let you learn how to operate it.

User Manual

CNC Manager

CNT830

____ Milling Controller

Drilling

Tapping (Option)

Boring (Option)

Contouring (Shape Milling)

Pocketing

3D CAM NETWORK PERFORMED

LINK FILE

G CODE -EDITOR

★ CONTENTS

• **FEATURES** 1-2

1. OPERATION PANEL 1-2

2. MAIN FUNCTIONS 1-5

3. COORDINATE SYSTEM 1-6

4. HOME 1-8

5. REMOTE CONTORL BOX 1-13

6. MANUAL 1-14

7. DRILL 1-21

8. TAP 1-27

9. BORE 1-33

10. CONTOUR 1-40

11. POCKET 1-51

12. GEOMETRY HELP FUNCTION 1-66

13. 3D CAM 1-123

14. LINK FILE 1-127

15. G CODE EDIT 1-130

16. BATHCH FILE 1-137

APPENDIX 一. SYSTEM PARAMETER 1-139

APPENDIX 二. ARM ATC OPERATION 1-153

FEATURES

Easy to operate : Only use these four graphic keys to do the complex machining without memorization of G and M codes.

Drilling : Normal drilling, pecking drilling, and chip break drilling.

Tapping : Rigid tapping and reciprocating tapping.

Boring : Arrival depth automatic tool retract.

Contouring : Cutter can be compensated from left or right side, taper.

Pocketing : Island avoidance, spiral, taper.

3D CAM : Apply for three dimension mould engraving.

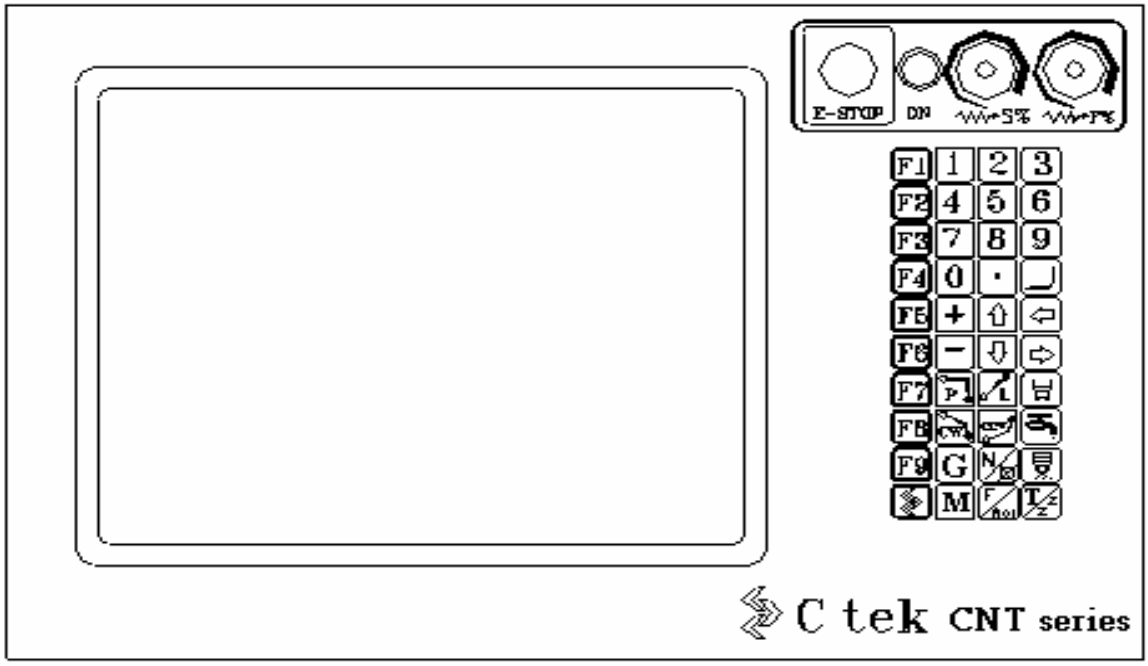
G GODE EDIT : Apply for ISO G code edit.

Repetitive : Apply for multi holes / position machining.

Tool path emulation : Double check the exact cutting path before machining.

Remote control box : Handwheel makes the alignment of the mould easily.

1. OPERATION PANEL



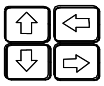
INTRODUCTION：

F1 TO F9 ：Function keys for different functions in different screens

0 TO 9 ：Number keys

. ：Decimal key

+ AND − ：Symbol keys

 ：Direction keys for moving cursor

：Input key for data input into memory

：Setting key for cutting point or starting point

：Setting key for straight line

：Clockwise radius setting key

：Counterclockwise radius setting key

：Set the movement for coolant state in Handle.

To control starting & stop of coolant state during execution.

：Set the movement for Air State in Handle.

To control starting & stop of air state during execution.

：T Code Key (while G Code editing)

To Control Starting & Stop of Sleep switch during execute.

: Quit key, go back to last page or stop.

: G Code Key (while G code editing)

: F Code key (while G code editing),
To control starting & stop of M01 (pause stop) during execution

: M Code Key (while G Code editing)

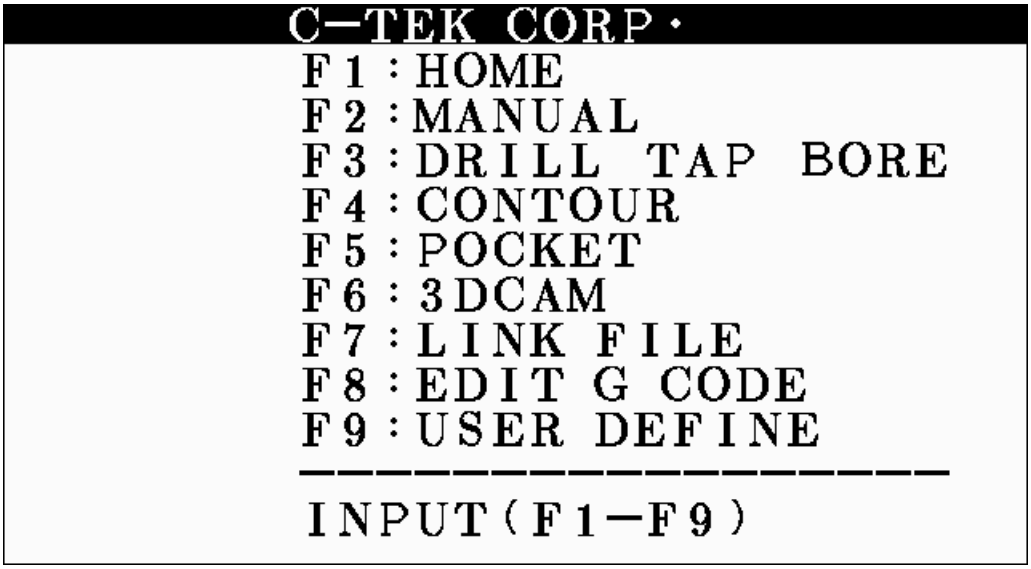
: N Code key (while G code editing).
To control of clear display during simulation or execution.

：Emergency stop

：Power switch

: Adjusting knob of spindle speed.

2. MAIN FUNCTIONS



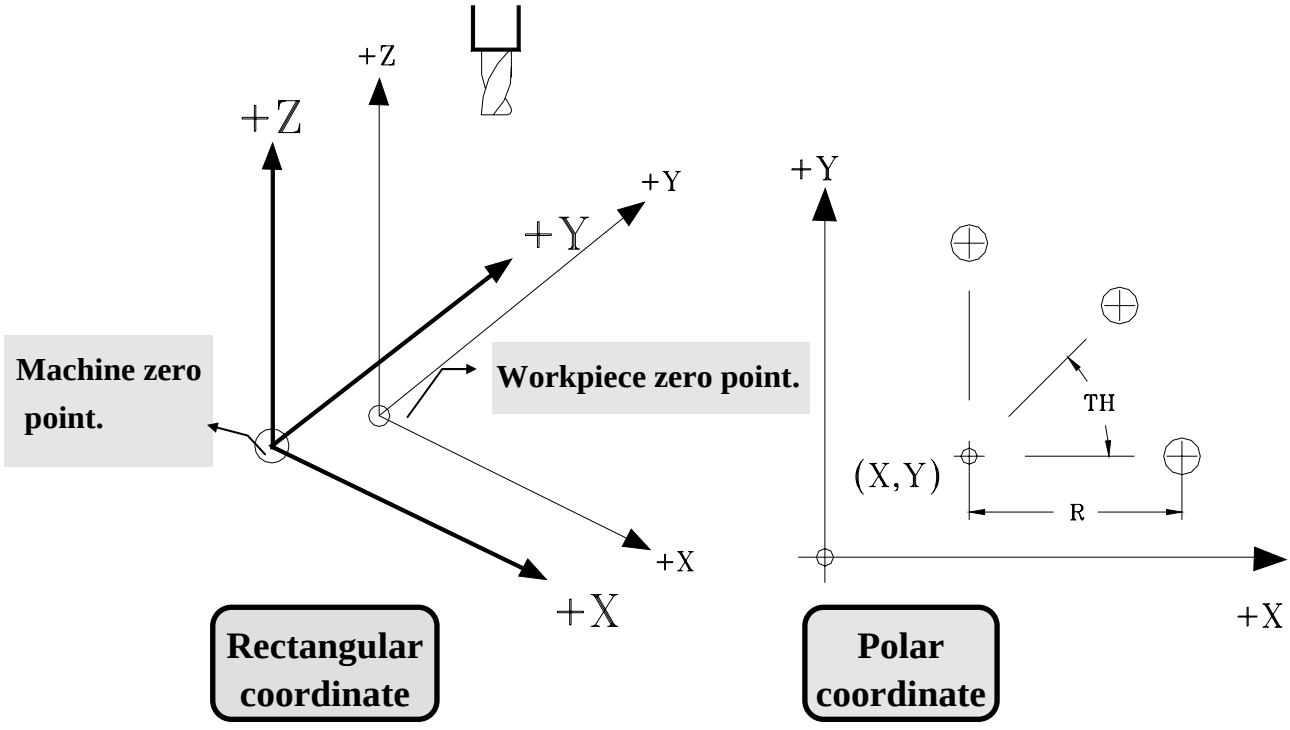
MAIN FUNCTIONS CHART

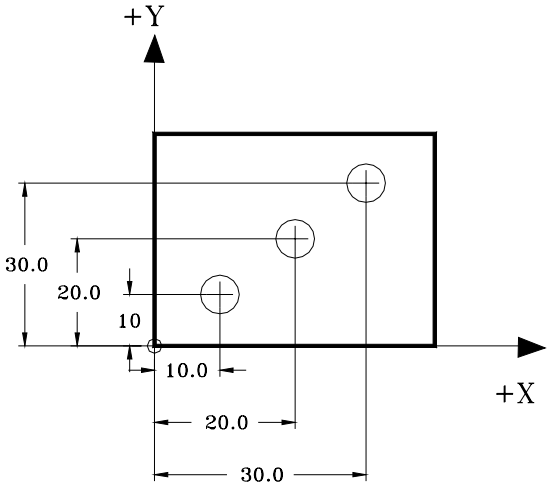
Under the main function menu, there is sub-functions chart.

	F1 HOME	F2 MANUAL	F3 DRILL TAP BORE	F4 CONTOUR	F5 POCKET	F6 3D CAM	F7 LINK FILE	F8 EDIT G CODE	F9 BAT
F1	X HOME	X SET	EDIT			Load File	EDIT		1.BAT
F2	Y HOME	Y SET	SIMULATE						2.BAT
F3	Z HOME	Z SET	EXECUTION						3.BAT
F4	XYZ HOME	XYZ MOVE	DELETE FILE						4.BAT
F5	LOAD ZERO	1/2 X	LIST FILE						5.BAT
F6		1/2 Y	COPY DRILL	COPY POCKET	COPY CONTOUR	TRANS FILE	Copy Link File	Copy G Code	6.BAT
F7			BACK UP	COPY CONTOUR	COPY POCKET		BACK UP		7.BAT
F8		G54-59 edit		BACK UP					8.BAT
F9		ATC							9.BAT
⇦	QUIT								

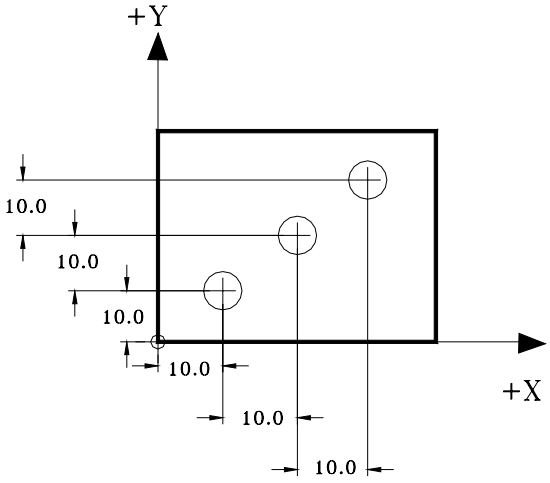
3. COORDINATE SYSTEM

1. **Rectangular coordinate system :** It consists of three coordinate axis (X,Y,Z) where the coordinate axis is perpendicular each other. The cross point of these three axes is called original point or zero point.
2. **Axis direction :** Cutter direction is the direction of axis.
3. **Machine zero point :** The zero point after work table Home returned.
4. **Workpiece zero point :** It is called "machining zero point". It's obtained after file loaded or mould aligned.
5. **Absolute coordinate :** An indicative way that the workpiece zero point is the reference base of machining size.
6. **Incremental coordinates :** Previous machined size is taken as the zero point for the existing machining size on the workpiece.
7. **Machine coordinate:** Machine zero point is the reference base of machining size.
8. **Polar coordinate :** Take any point (X, Y) as an original point and the machined size is represented by radius (R) and angle (TH).



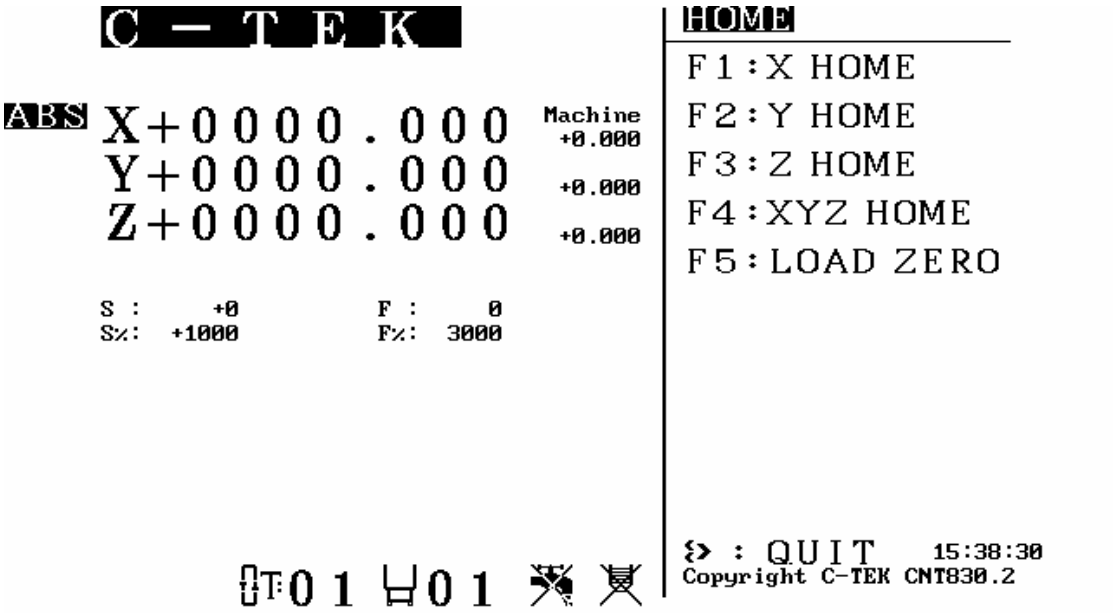


Absolute coordinate



Incremental coordinate

4. HOME



Time to use HOME RETURN:

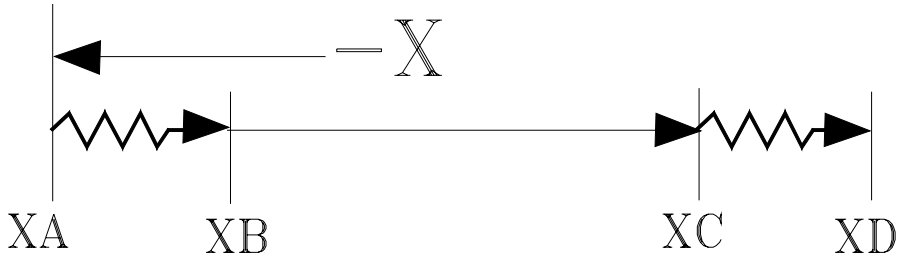
1. Execute home return first when the system power on, i.e. execute machine zero point.
2. After the three axes home return, press F5 to load the memory with last mould's zero point coordinate.

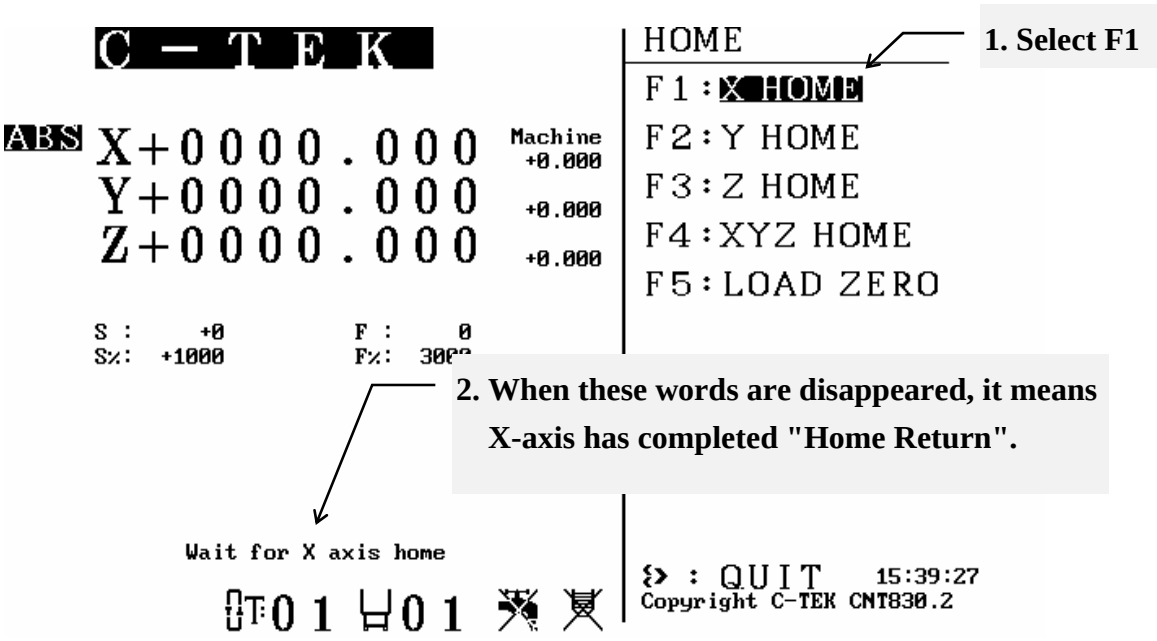
Caution :

1. Home return must be done whenever the power is on. This is to make sure the program has a fixed reference as a zero point.
2. In order to prevent mould collision, single axis home return is recommended, i.e. after one axis is home returned, then perform another axis.
3. It is designed that only the last mould zero point saved in the controller. When you have set the x, y, z coordinate, last zero point coordinates have been replaced. That is, you can't find the old zero point coordinates out anymore.

F1: X-axis home return
Description:

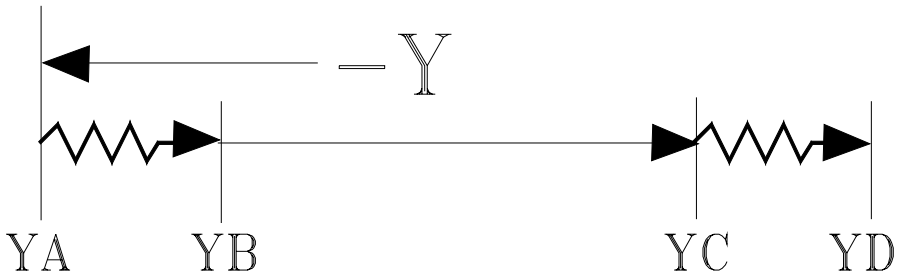
1. X-axis moves rapidly toward -X direction, until the limit block touches - X limit switch. (System set Negative home return)
2. X-axis moves slowly toward +X direction, until the limit block escapes -X limit switch.
3. X-axis moves rapidly toward XC point again. (system parameter HOME MOVE X)
4. When XC point is reached, X-axis moves slowly toward +X direction until the XD point on Encoder is found.
5. The distance between XC point and XD point is 1/2 round of motor.

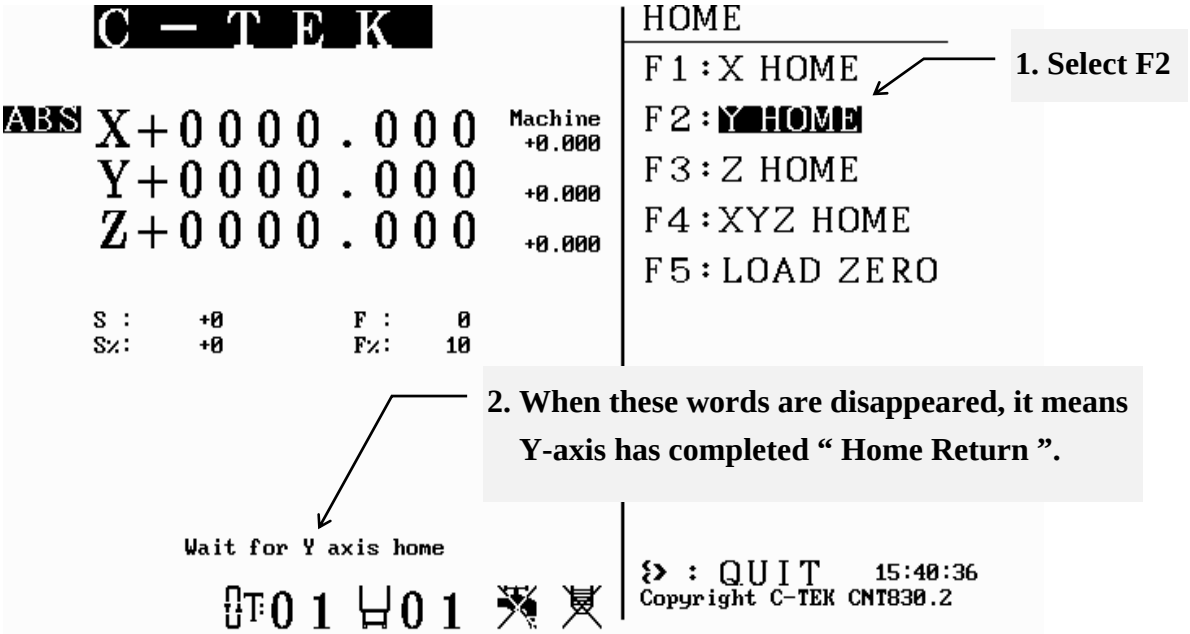




F2: Y-axis home return
Description:

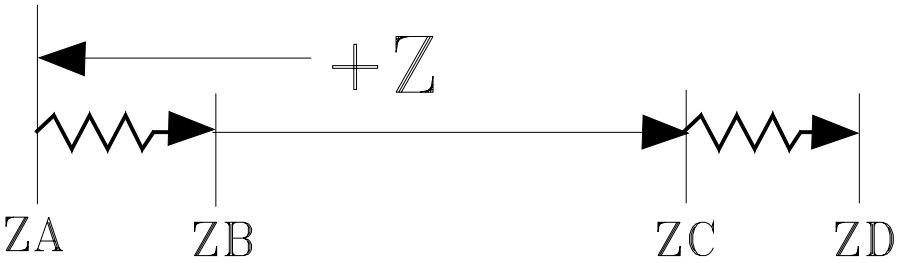
1. Y-axis moves rapidly toward -Y direction, until the limit block touches -Y limit switch. (System set Negative home return)
2. Y-axis moves slowly toward +Y direction, until the limit block escapes -Y limit switch.
3. Y-axis moves rapidly toward YC point again.
4. When YC point is reached, Y-axis moves slowly toward +Y direction until the YD point on Encoder is found.
5. The distance between YC point and YD point is 1/2 round of motor.

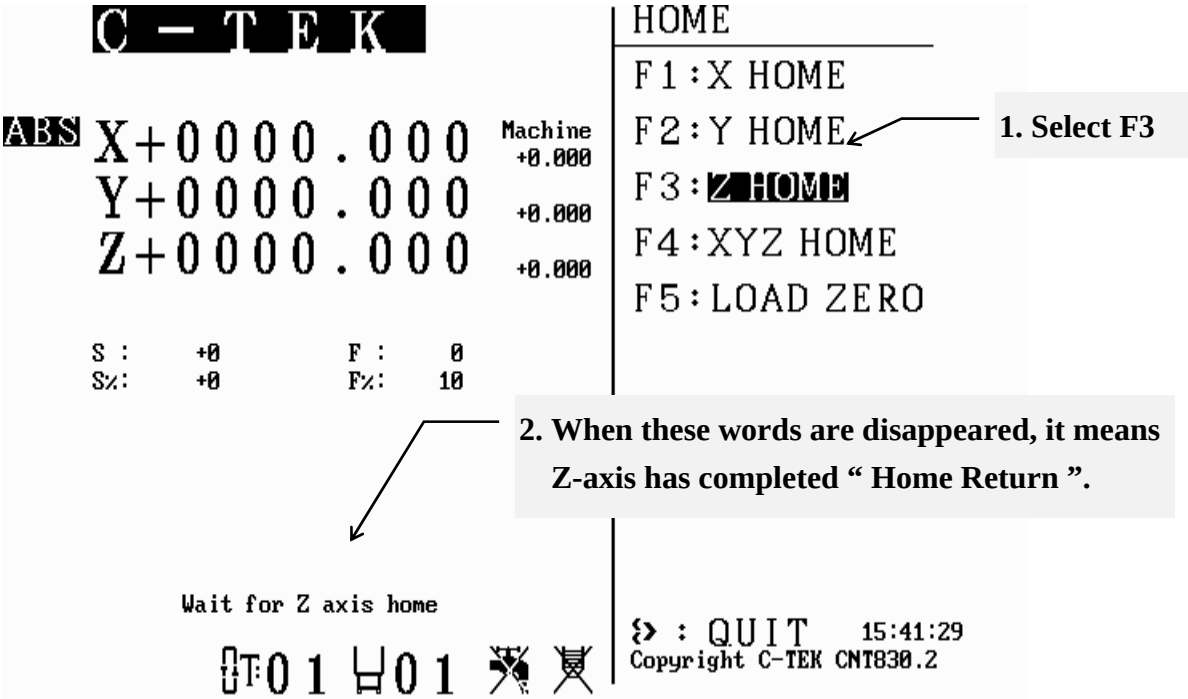




F3: Z-axis home return
Description:

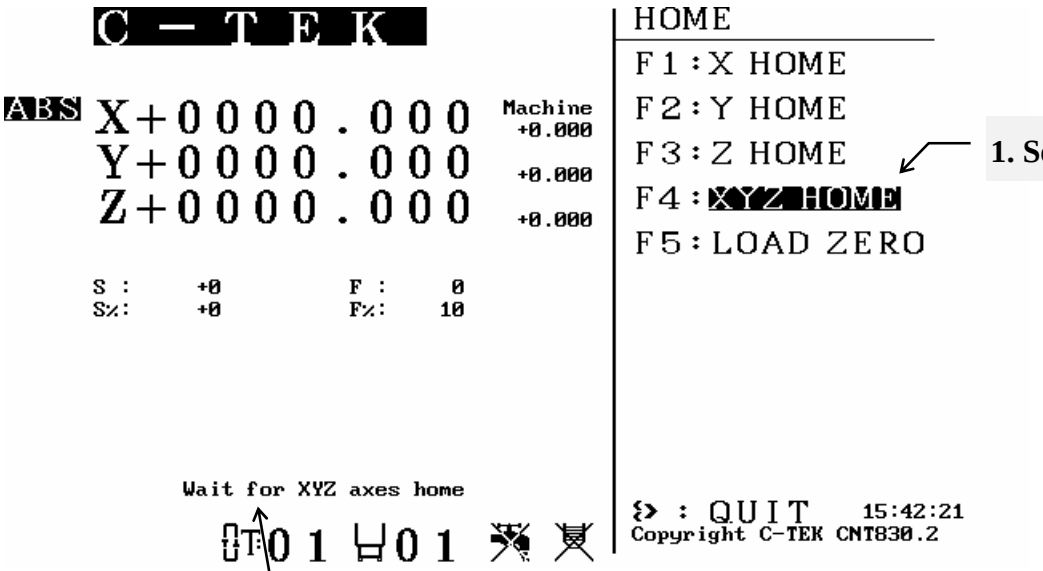
1. Z-axis moves rapidly toward +Z direction until the limit block touch +Z limit switch. (System set Positive Home return)
2. Z-axis moves slowly toward -Z direction until the limit block escapes +Z limit
3. Z-axis moves rapidly toward ZC point again.
4. When ZC point is reached, Z-axis moves slowly toward -Z direction until the ZD point on Encoder is found.
5. The distance between ZC point and ZD point is 1/2 round of motor.





F4: XYZ axes zero return
Description:

Three axes zero return simultaneously as stated above in F1, F2, and F3.

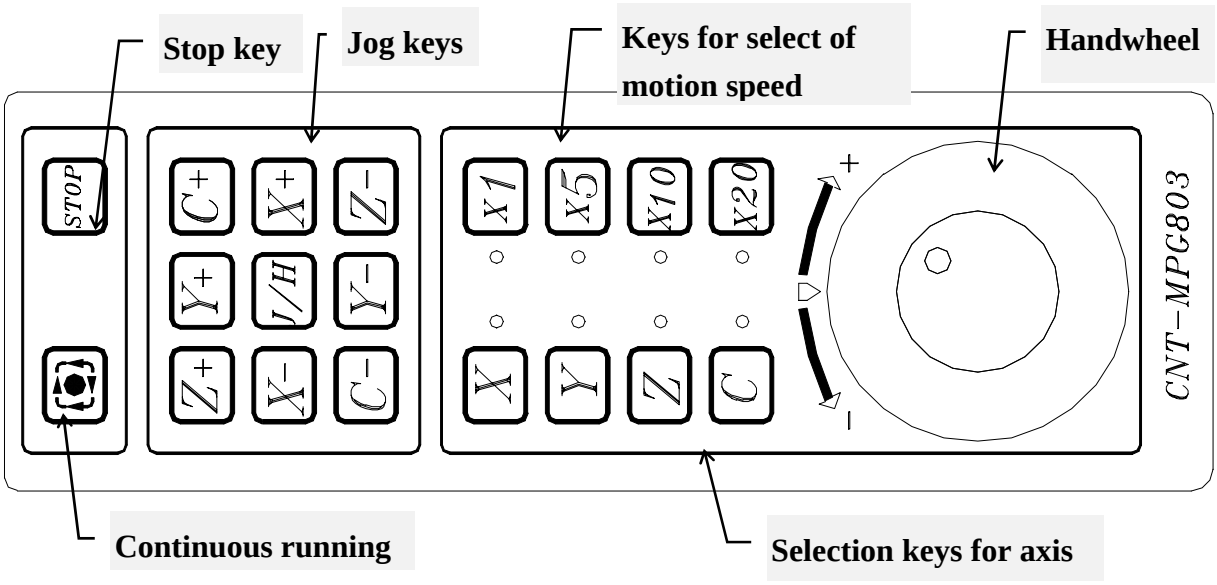


F5: Load zero

Description:

When you press F5, the Zero point of last coordinate would be load in the memory again.

5. REMOTE CONTROL BOX



REMOTE CONTROL PANEL

Caution:

Remote control box is only accessible in 『HANDLE』. However, it can also be accessed when 『STOP』key is used during machining and『Worktable home return』.

Introduction:

Continuous running key: Continue machining after pause key is pressed.

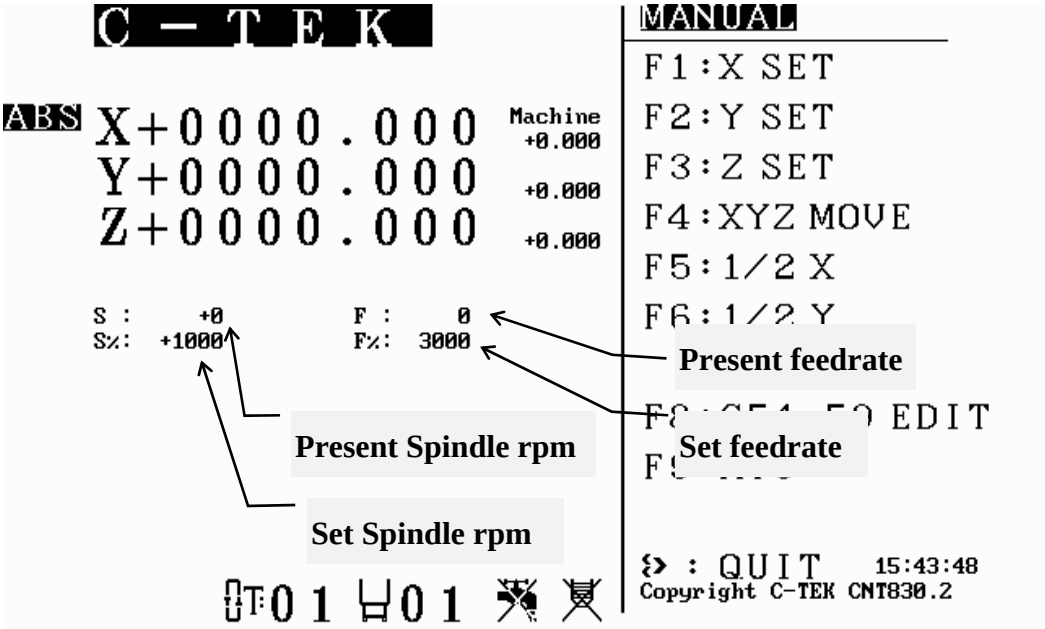
Stop key: Stop a moving worktable.

JOG keys: Press any Jog key and the worktable will be activated toward the direction dedicated by the JOG key.

Selection keys for axis direction: When an axis is selected, the indication lamp will be activated.

Handwheel: Turning handwheel clockwise is positive (+) direction while counterclockwise is negative (-) direction.

6. MANUAL



Time to use it:

1. Mould alignment: The『MANUAL』and『HANDWHEEL』 key in the remote control box are the ideal functions for mould alignment.

2. Zero setting for mould: Machining zero point can be set by the functions of 『X SET』,『Y SET』,『Z SET』,『1/2X』, and『1/2Y』. After you set it, it will be saved in the hard disk, and can be reload the zero point when the power fails.

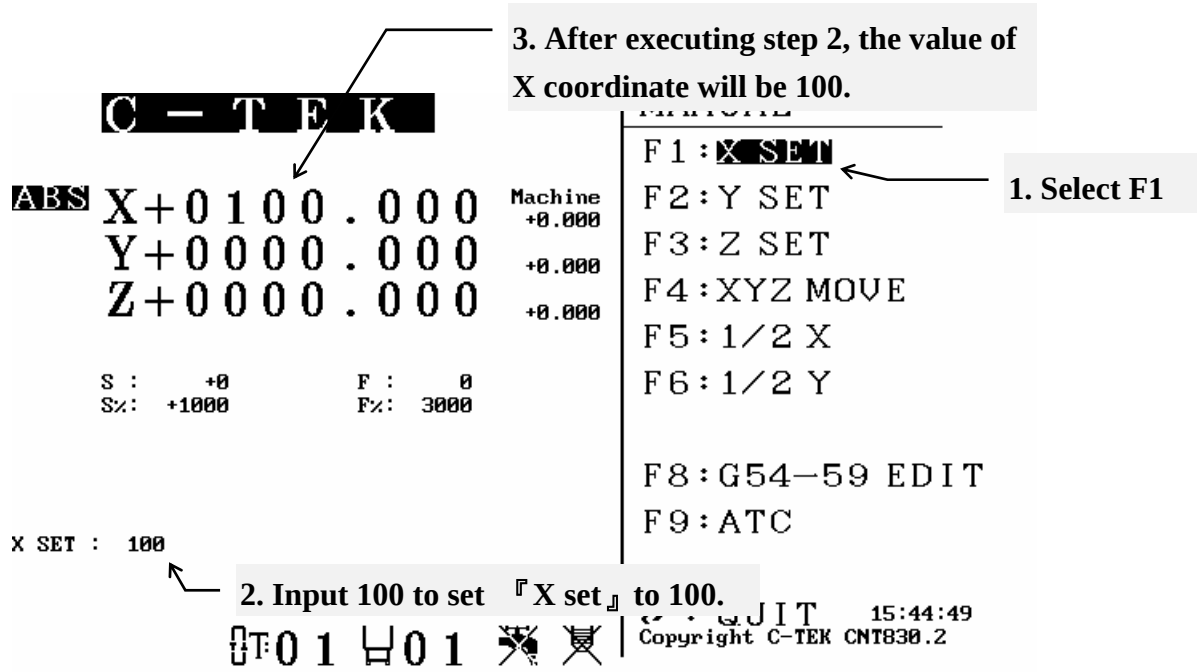
3. Manual: Use 『Handwheel』in the remote control box to perform manual machining. It is easier and faster to do machining when mould alignment. Or adjust the panel F% switch to control the XYZ move speed, and use the remote control box X+X-Y+Y-Z+Z- key, for setting Manual Speed.

4. Positioning : For those known coordinate positions,『XYZ MOVE』can be used as JOG in order to reach accurate positioning.

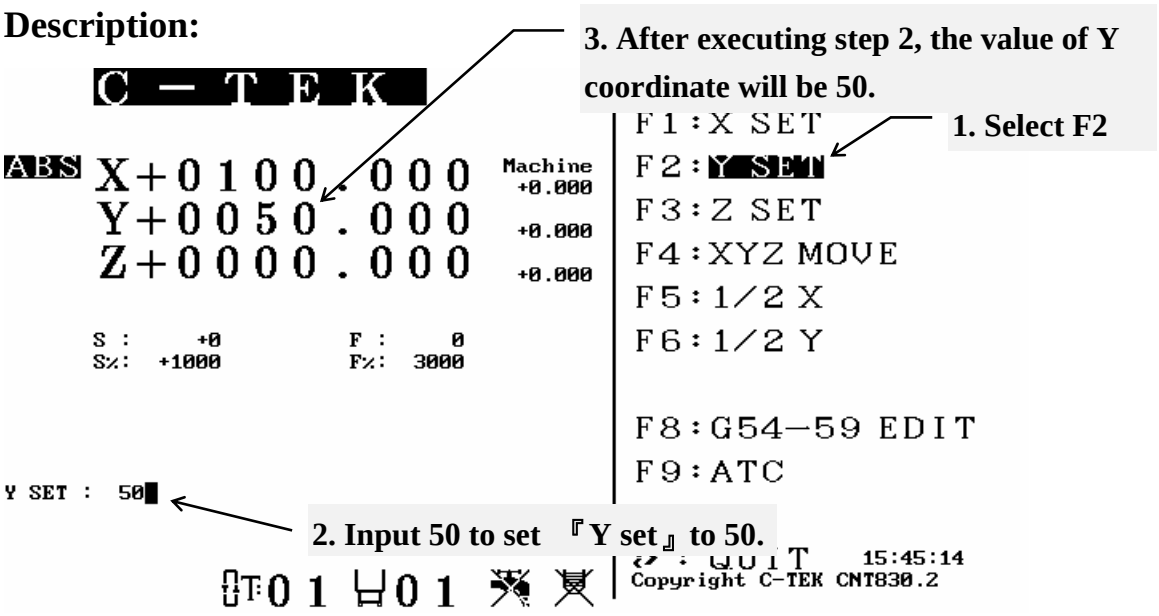
Caution:

1. If XYZ movement is performing, to stop movement emergency uses the 『 < > : Quit 』 or 『Stop』 key on remote control box.
2. The input value has a certain range for『X SET』,『Y SET』and『Z SET』. The range got something to do with the axis stroke and built-in parameters. Any value exceeding the range will not be accepted.

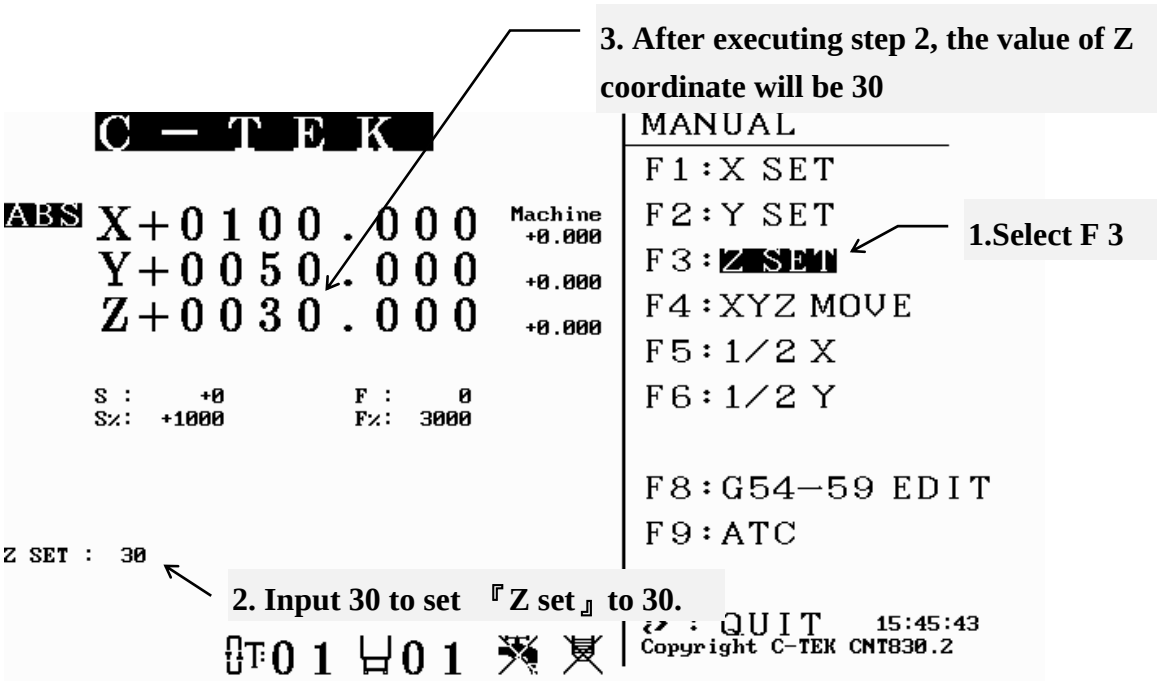
F1: X SET
Description:



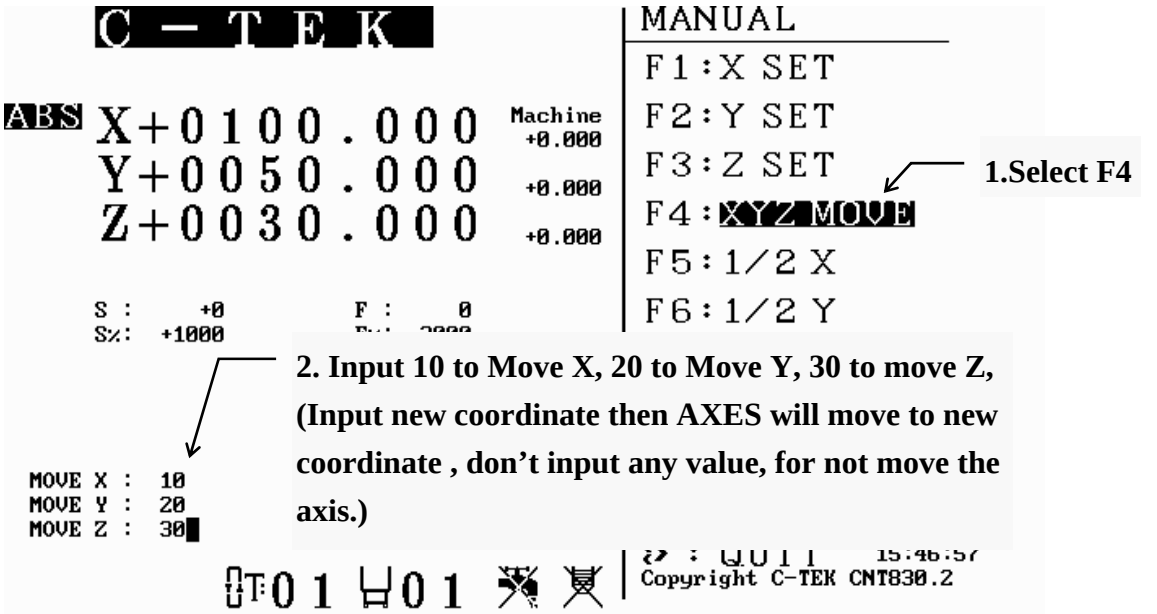
F2: Y SET
Description:



F3: Z SET
Description:

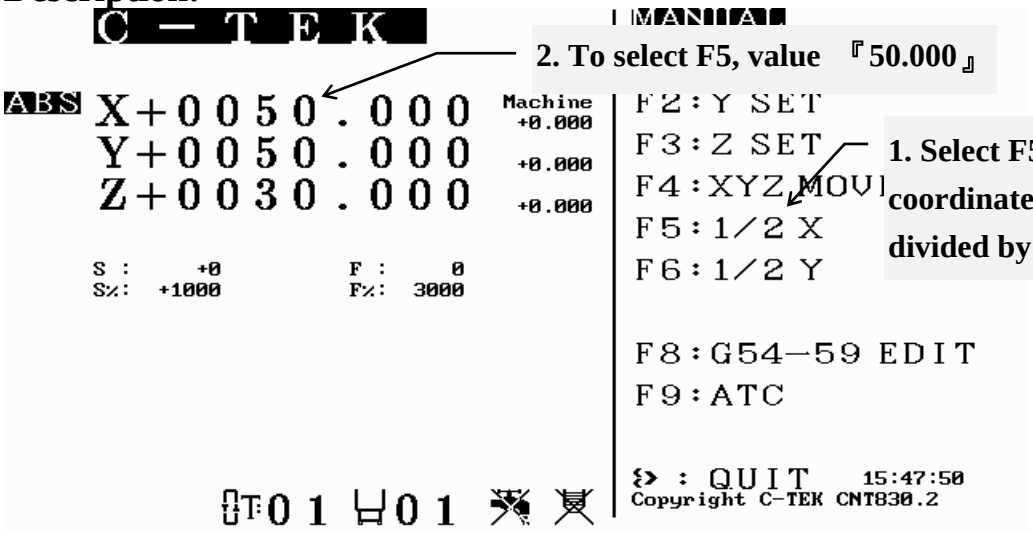


F4: XYZ MOVE
Description:


C — T E K
MANUAL
F1:X SET
F2:Y SET
F3:Z SET
F4:XYZ MOVE
F5:1/2 X
F6:1/2 Y
ABS X+0100.000 Machine +0.000
Y+0050.000 +0.000
Z+0030.000 +0.000
S : +0 F : 0
S%: +1000
1.Select F4
MOVE X : 10
MOVE Y : 20
MOVE Z : 30
2. Input 10 to Move X, 20 to Move Y, 30 to move Z, (Input new coordinate then AXES will move to new coordinate , don't input any value, for not move the axis.)
T01 H01
Copyright C-TEK CNT830.2

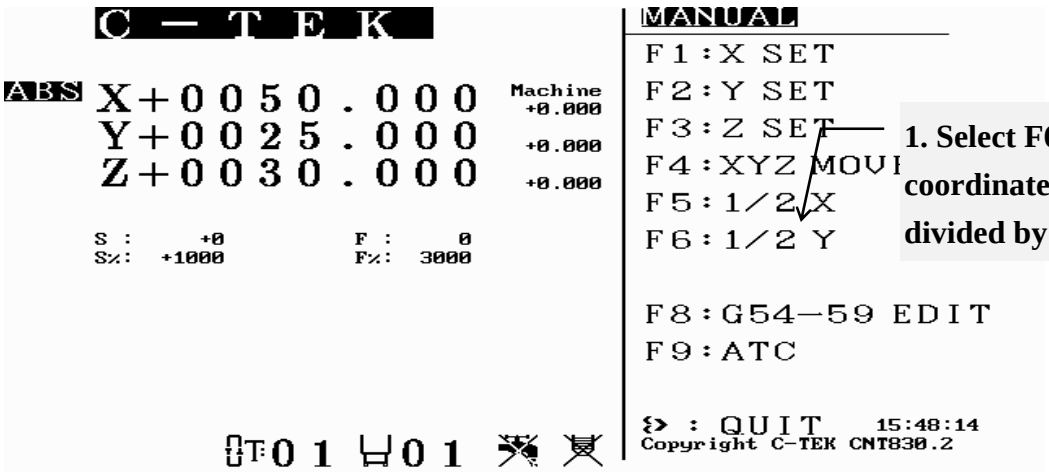
F5: 1/2 X

Description:



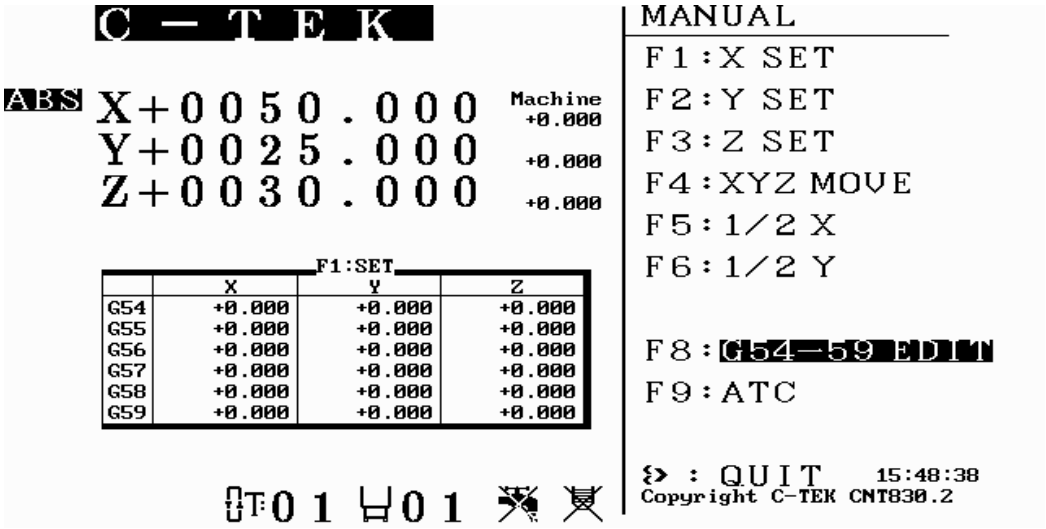
F6: 1/2 Y

Description:

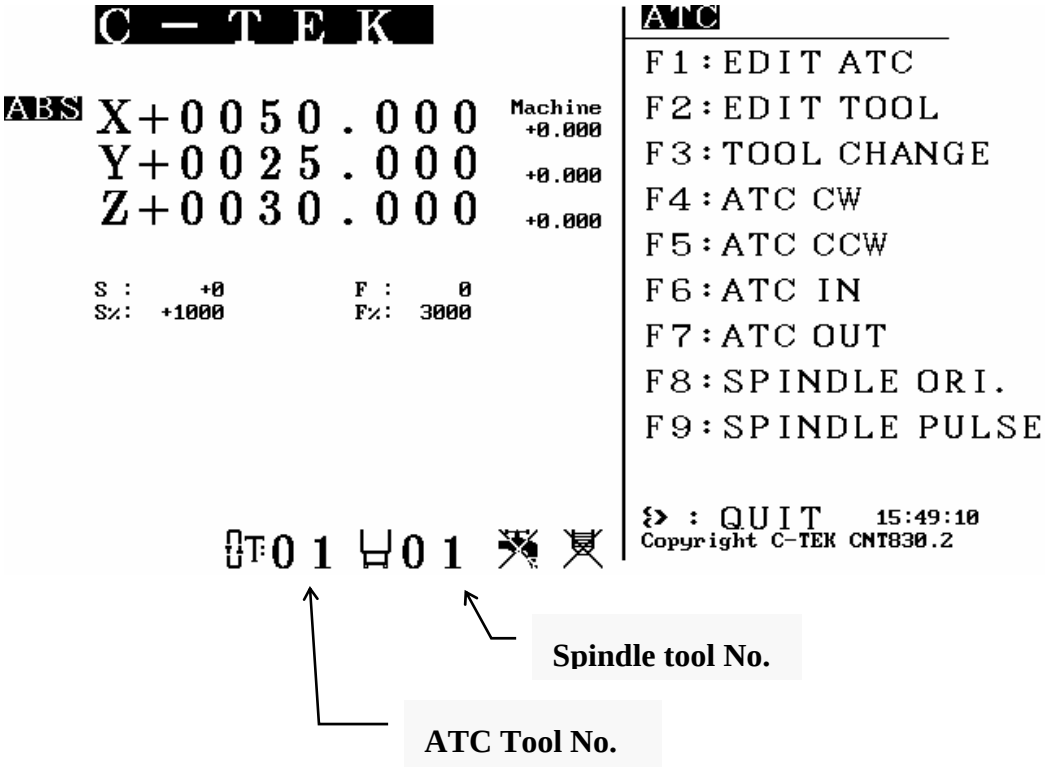


F8:G54-59 Edit

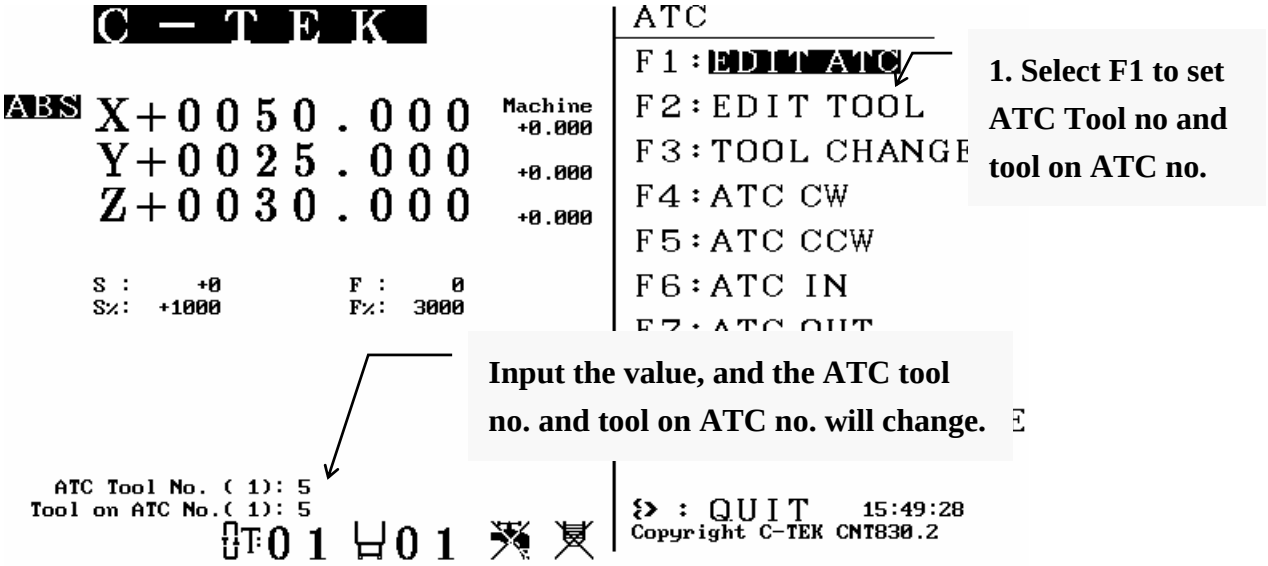
Description:



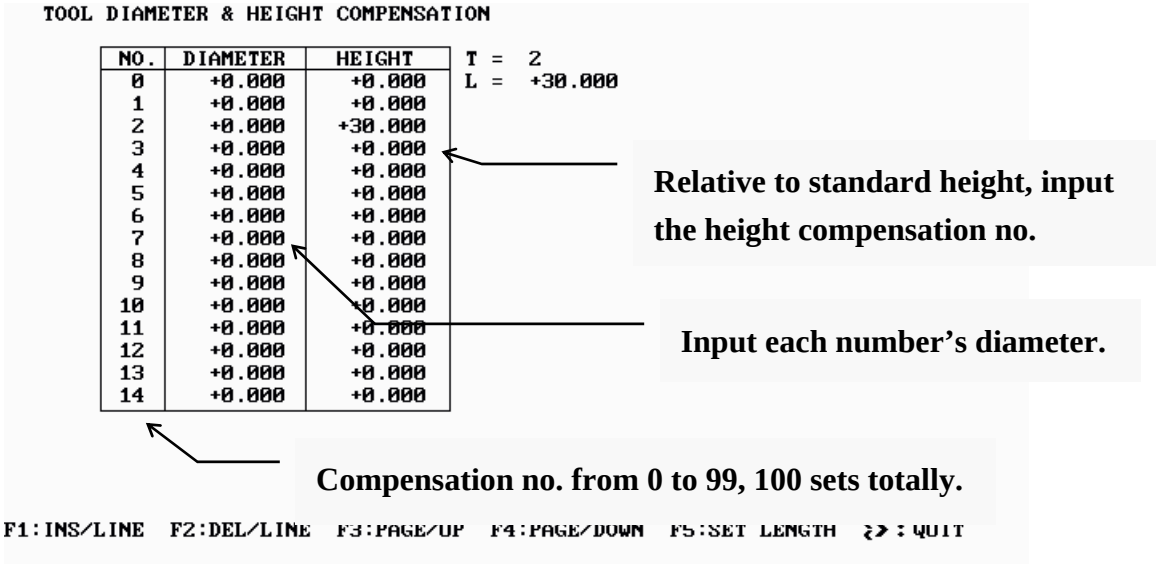
F9: ATC (Option)
Description:



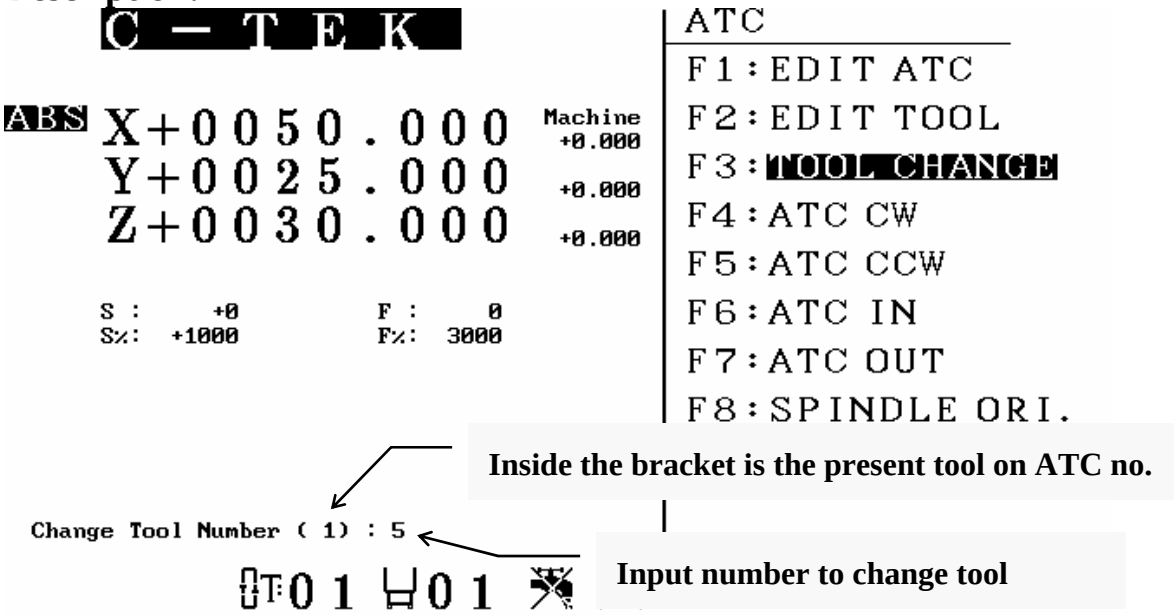
F1: EDIT ATC
Description:



F2: EDIT TOOL
Description:



F3: TOOL CHANGE
Description:



F4: ATC CW: When you press the key, ATC turns clockwise, i.e. if the ATC Tool no. is 5 which is relative to TOOL ON ATC NO., the number turns into 6 after you press the key.

F5: ATC CCW: When you press the key, ATC turns anti-clockwise, i.e. if the ATC TOOL NO. is 5 which is relative to TOOL ON ATC NO., the number turns into 4 after you press the key.

F6: ATC IN: Press the key to push ATC toward spindle direction until inside Limit switch is touched.

F7: ATC OUT: Press the key to push ATC OUT from spindle direction until Side limit switch is touched.

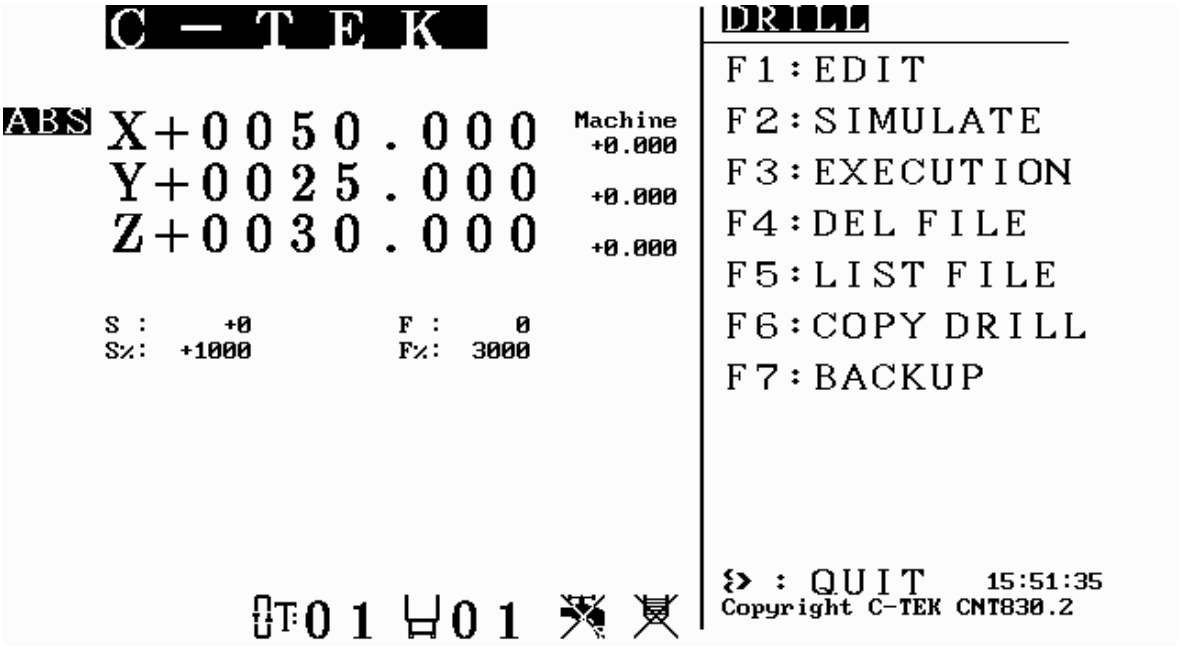
F8: SPINDLE ORI: Press the key to turn the spindle into the TOOL CHANGE ORIENTAL point.

F9: SPINDLE PULSE: Press the key and you can turn the spindle by your hand. The video displays an image of relative pulse value. But It is only used when computer trial.

CAUTION:

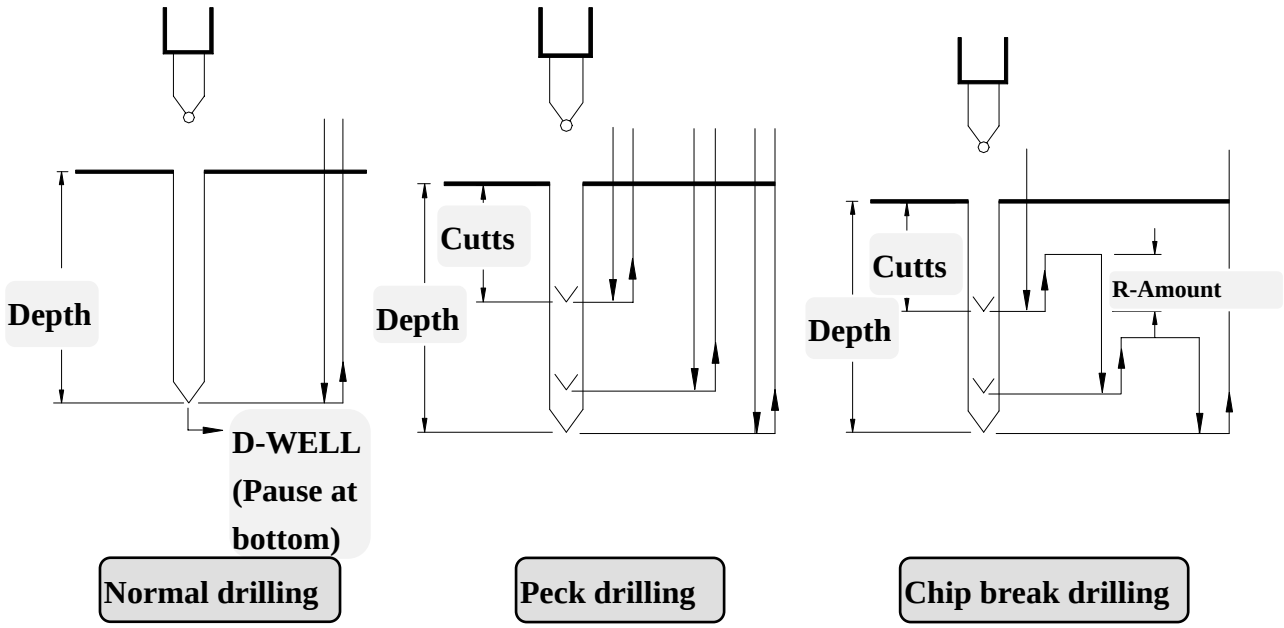
1. **When the z-axis' position is lower than TOOL CHANGE'S position, it's no use to press "F6: ATC IN".**
2. **Please don't perform TOOL CHANGE function while ACT IN.**
3. **The usage of tool height compensation:**
   a. **Go to HANDLE and put the standard tool into the spindle. Move Z axis to make the nose of the tool touching the standard surface. Press F3 and set z axis 0. If the standard tool is one tool of the ATC, the tool height compensation which is according to this tool should be set 0.**
   b. **Use the function of TOOL CHANGE to put the tools into the spindle in turn.**
   c. **Move Z axis to make the nose of the tool touching the standard surface. Input Z axis value to the tool height compensation which is according to the tool without performing ZERO SET.**
   d. **Repeat item b and c until all the tools you want to use were set.**
   e. **If the tool you want to use is longer than the standard tool, the value you set is "+". On the contrary the value is "-".**

7. DRILL

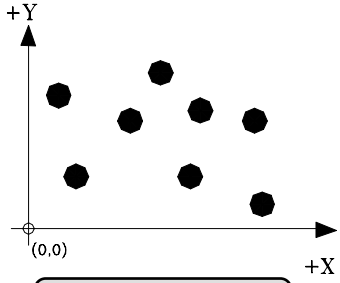


Function description:

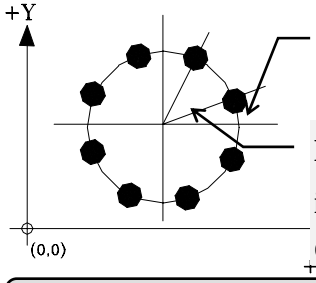
1. **Pattern of drilling:** 『Normal drilling』, 『Peck drilling』, and 『Chip break drilling』.



2. Setting of hole position: 『Random position』, 『Right angles & equi-distant position』, 『Slope & equidistant position』, 『Radius & equiangular position』. These four different ways can be mixed and combined to be operated.



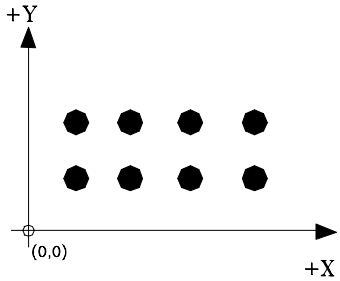
Random position



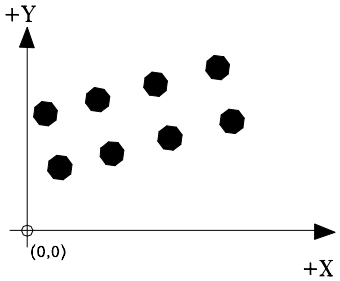
STH is the angle between the first hole and zero degree.

DTH is『Rectangle』 degree, it means each hole is rectangle distance。

Radius & equiangular position



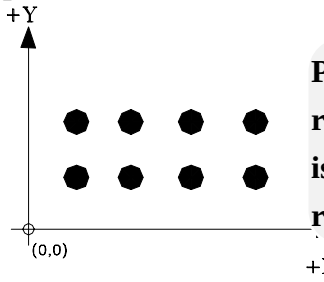
Right angle & equidistant position



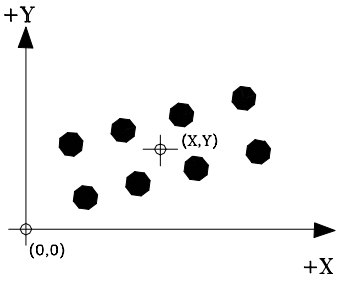
Slope & equidistant position

Caution : The 『Random position』 can be set by using of absolute coordinate.

3. Rotation X, Y (Rotating angle)**:** Angle rotation can be done at any coordinate point dedicated。



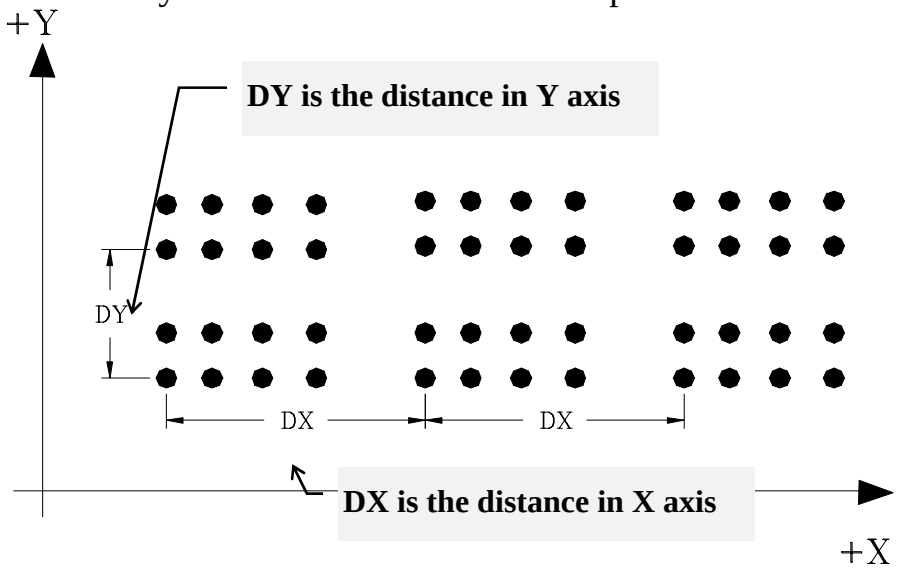
Picture shown right-hand side is after angle rotating.



4. D-Well (Pause at bottom)**:** When drilling to the desire depth, spindle remains at the bottom to enable the hole bottom shining.

5. TOOL NO.: You have to input TOOL NO. when you want to use ATC TOOL in machining. However, if you input 0 it won't change the tool, but use the tool on the spindle to perform the function. In the case, it is no use to input diameter and height compensation no. The system will select the first parameter's diameter value to do diameter compensation and select tool zero point to do depth control.

6. Nx、Ny (Receptive machining)**: D**uplicate previous machining sequence in an exactly same sizes in the same workpiece.



Caution: The minimum setting values of Nx and Ny are 1 and 1, respectively, as Picture above Nx=3 and Ny=2.

F1: EDIT

Description :

EDIT PROGRAM NO.: 1 (1) ABS(TIMES=0),INC(TIMES=OTHER'S)

STEP	X	Y	Z	R	DTH	STH	TIMES
1	+10.000	+0.000	+5.000	+0.000	+0.000	+0.000	4
2	-10.000	+10.000	+4.000	+0.000	+0.000	+0.000	4
3	+40.000	+30.000	+3.000	+0.000	+0.000	+0.000	0
E 4	+0.000	+0.000	+2.000	+10.000	+45.000	+20.000	8

Press quit key to switch the display to next page " Parameter setting ".

F1:INS/LINE F2:DEL/LINE F3:PAGE/UP F4:PAGE/DOWN F5:DATA/END {>:QUIT

Edit description :

EDIT PROGRAM NO : Input program number.（The number between 0 to 999）.
Last program number in the brackets

Z : Z value is the starting coordinate value of " machining depth". It is always represented by absolute coordinate.

R : When R≠0, it's setting as "Radius & equiangular position". The X and Y are the center of the arc.

Times : When TIMES=0, the holes in the column are set to absolute coordinate.
When TIMES is not 0, the holes in the column are set to increment coordinate.

F1:INS/LINE : To insert a column. **F2:DEL/LINE :** To delete a column.

F3:PAGE/UP : Page up. **F4:PAGE/DOWN :** Page down.

F5:DATA/END : "Cursor" moves to any position, then press F5 to set ending point in the column, the point is represented by " E ".

Parameter setting :

```
 EDIT PROGRAM NO.: 1     (1  )     ABS(TIMES=0),INC(TIMES=OTHER'S)

Tool Diameter          :     5.000      Tool No. (T)           :     1
Move Height            :    30.000      Tool Dia.   Compen No. :     1
Z Final Depth          :    10.000      Tool Length Compen No. :     1
Z Cutts                :     3.000      Spindle RPM            : +1000
Ret. (0:R,1:Sys.c)     :     1          C Axis DTH             :    +0.000
R height               :     3.000      C Axis STH             :    +0.000
D-Well                 :     1.0
*********(1)***********:     1
Rotation (X,Y)         :    +0.000
         ABS X         :    +0.000
             Y         :    +0.000
Nx:      2           Ny:     2
Dx:    +50.000       Dy:   +50.000
Feedrate               :  1000
Sleeping (0=Off,1=On)  :     0
Display :   0        **:     1
Mode (0=Auto,1=Manu)   :     0
Index No.              :     1
Image (0=Off,1=X,2=Y)  :     0
      Distance         :    +0.000
```

＊Press the QUIT key to go back DRILL screen.

Parameter description :

Tool Diameter : Set drill tool diameter.

Move Height : When the hole position is moved, in order to prevent collide of mould, the spindle will moves to the position of “Safe Height ” + “ Z axis starting coordinate”.

Ret. (0:R,1:Sys.c) : Retract tool to follow parameter of “R height” position or to the height registered by system parameter. Or to prevent “ Over travel ” caused by immediately stop (clearance).

R height : Retract tool position.

Feedrate : To adjust feeding rate, use『F% Feedrate adjusting knob』from 10% to 150%.

Sleeping : 1 is to activate『sleeping switch』.

Mode (0=Auto,1=Manu) : 0 is automatic movement plus drilling. 1 is automatic movement, then press the button of 『Continuous running』 on『Remote control box』for drilling action.

Spindle RPM : Input the spindle rpm

Tool No. (T) : Input tool No.

Tool Dia. Compn No. : Input diameter compensation no.

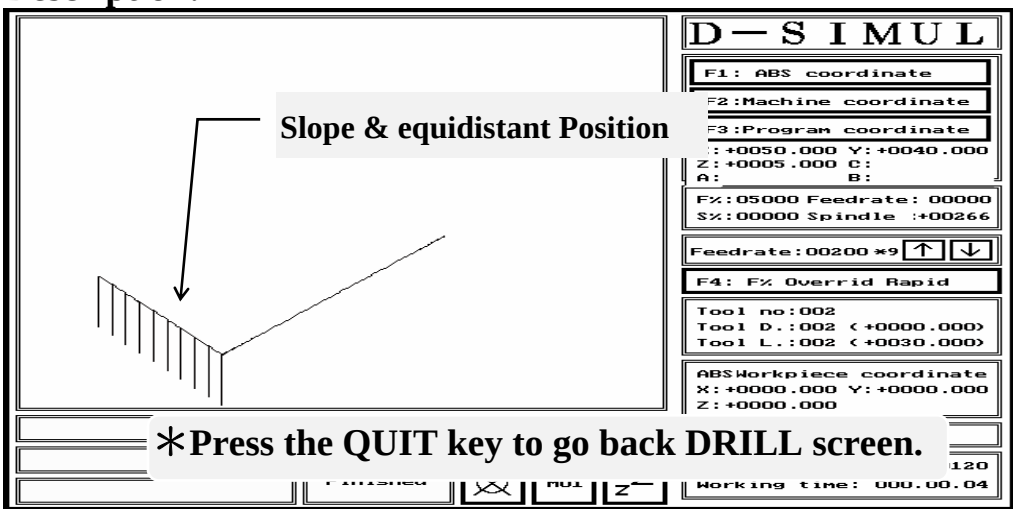
Tool length Compn No. : Input height compensation no.

C Axis Dth. : Incremental C axis angle. If non zero, the NY parameter is assumed to refer to the number of repeats in C.

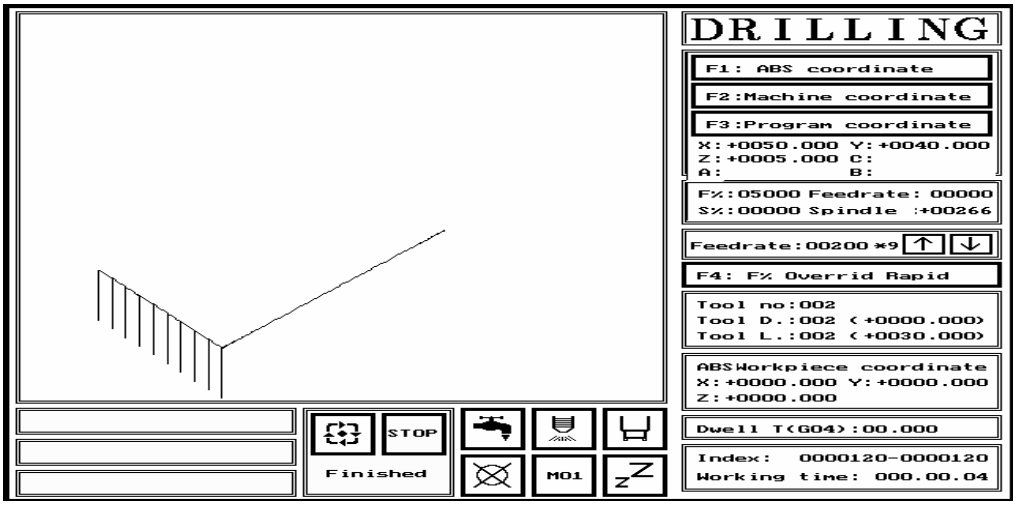
C Axis Sth. : Initial C axis angle.

F2: SIMULATION

Description:



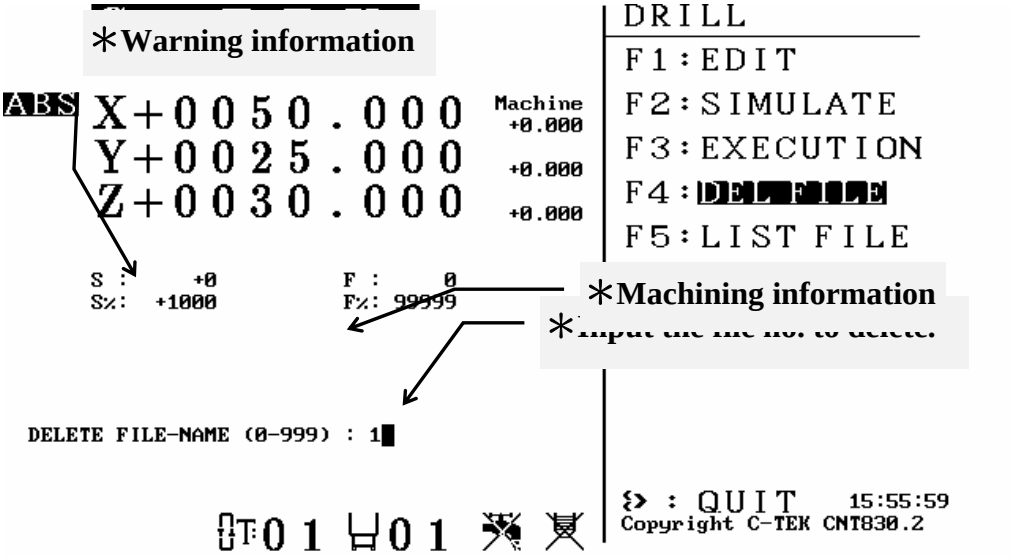
F3: EXECUTION



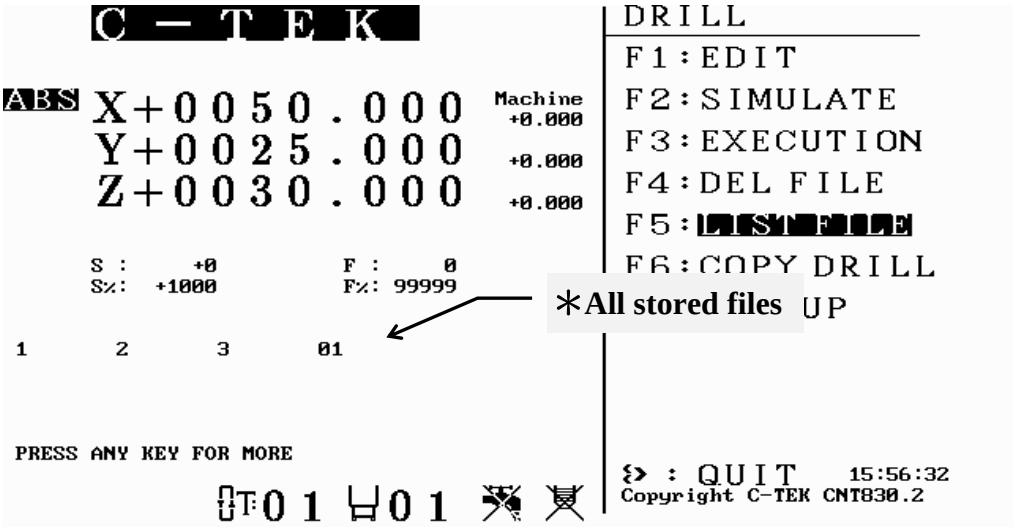
Description:

F4: DEL FILE

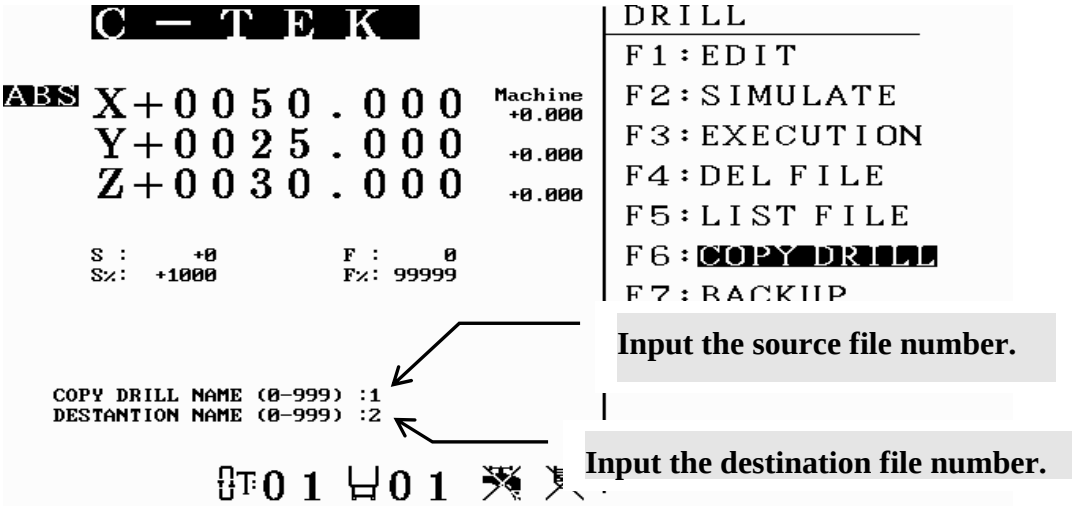
Description:



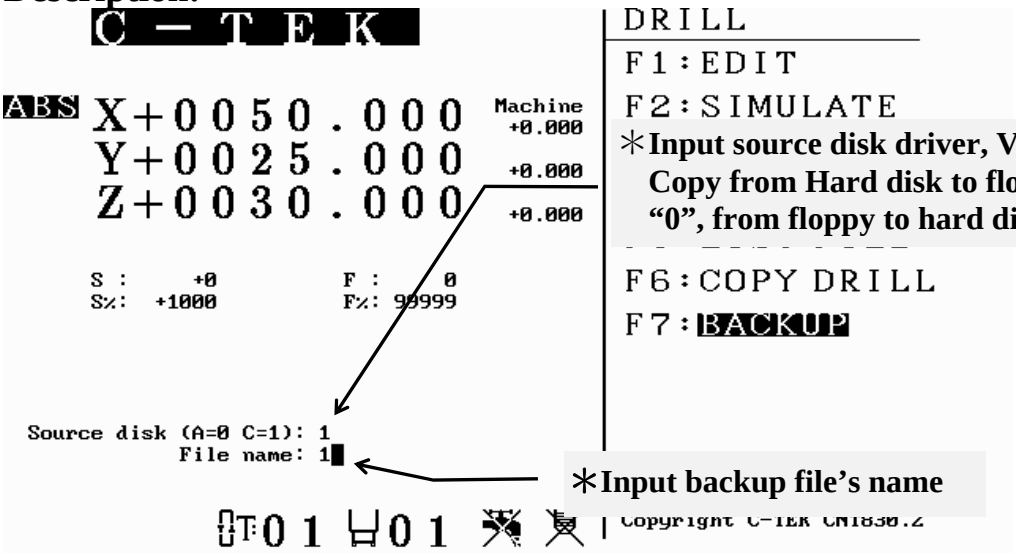
F5: LIST FILE
Description:



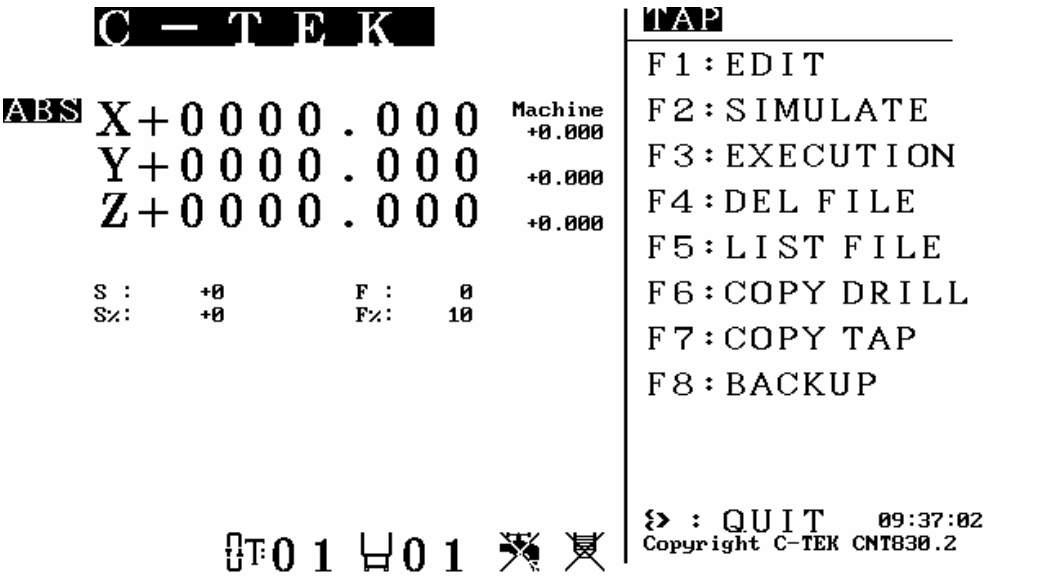
F6: COPY DRILL:
Description:



F7: BACKUP
Description:

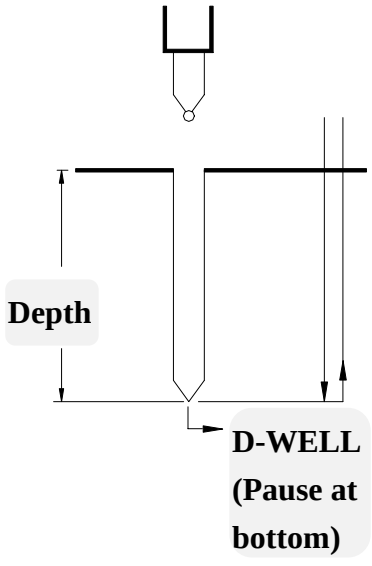


8. TAP (Option)

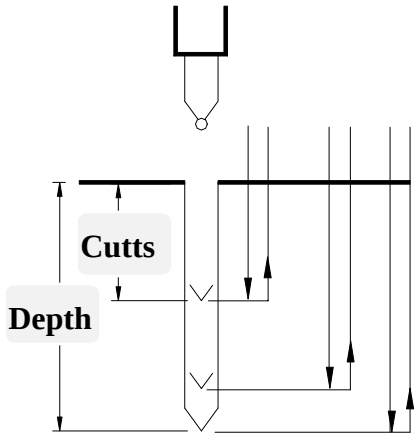


Function description:

1. Pattern of tapping: 『Normal tapping』 and 『Peck tapping』

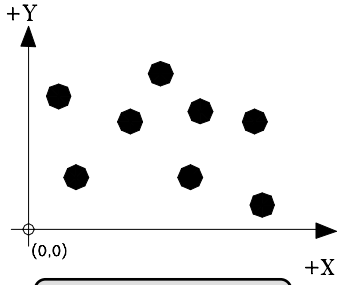


Normal tapping

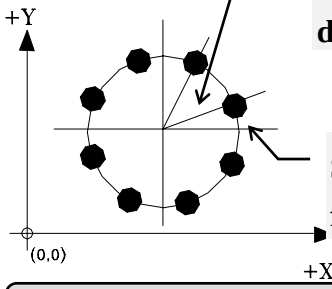


Peck tapping

2. Setting of hole position: 『Random position』, 『Right angles & equi-distant position』, 『Slope & equidistant position』, 『Radius & equiangular position』. These four different ways can be mixed and combined to be operated.



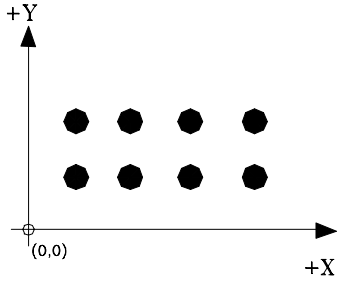
Random position



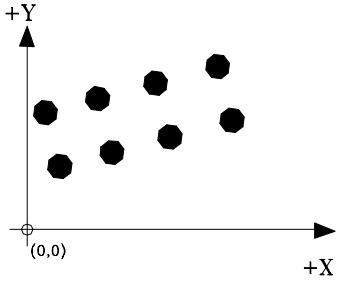
DTH is『Rectangle』 degree, it means each hole is rectangle distance。

STH is the angle between the first hole and zero degree.

Radius & equiangular position



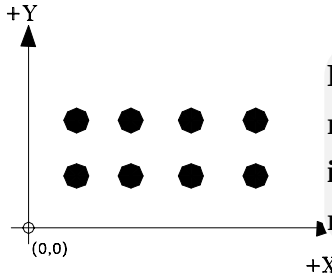
Right angle & equidistant position



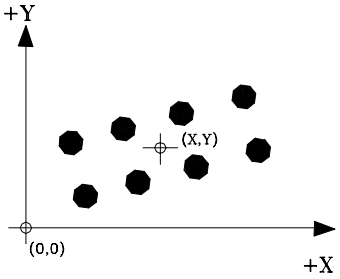
Slope & equidistant position

Caution : The 『Random position』 can be set by using of absolute coordinate.

3. Rotation X,Y(Rotating angle)**:** Angle rotation can be done at any coordinate point dedicated。



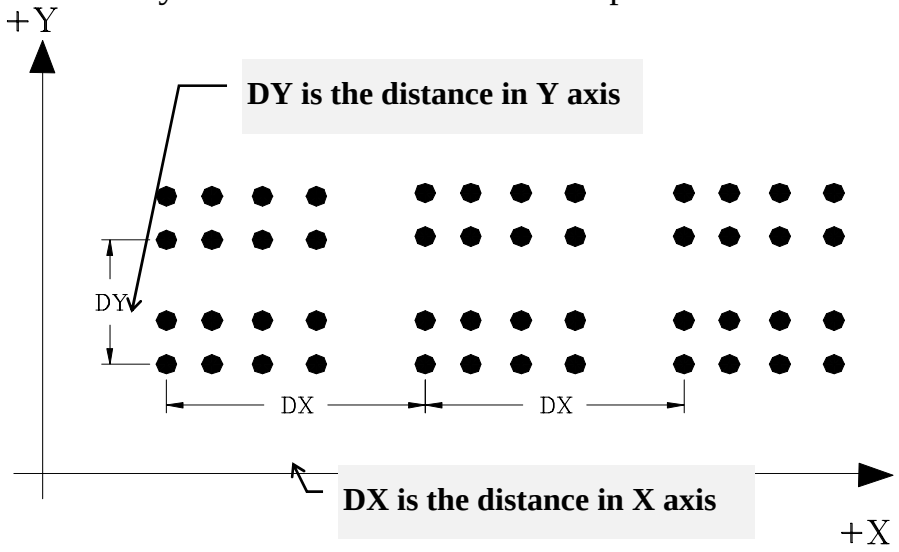
Picture shown right hand side is after angle rotating.



4. Index No.: For automatic return 『machining interrupt』 point.

Caution: For same reason you may be changed cuter when working piece was machining, you must press STOP key on remote box and record the Index No. by writing down.

5. Nx、Ny (Receptive machining）**:** To duplicate previous machining sequence in an exactly same sizes in the same workpiece.



Caution :：The minimum setting values of Nx and Ny are 1, respectively, as picture above Nx=3 and Ny=2.

F1: EDIT

Description :

EDIT PROGRAM NO.: 1 (1) ABS(TIMES=0),INC(TIMES=OTHER'S)

STEP	X	Y	Z	R	DTH	STH	TIMES
1	+37.000	+110.000	+0.000	+0.000	+0.000	+0.000	0
2	+87.500	+82.600	+0.000	+0.000	+0.000	+0.000	0
3	+87.500	+42.300	+0.000	+0.000	+0.000	+0.000	0
4	+37.000	+15.000	+0.000	+0.000	+0.000	+0.000	0
5	+203.000	+15.000	+0.000	+0.000	+0.000	+0.000	0
6	+120.550	+35.200	+0.000	+0.000	+0.000	+0.000	0
7	+203.000	+110.000	+0.000	+0.000	+0.000	+0.000	0
8	+157.500	+82.600	+0.000	+0.000	+0.000	+0.000	0
9	+230.000	+82.600	+0.000	+0.000	+0.000	+0.000	0
10	+119.000	+76.000	+0.000	+0.000	+0.000	+0.000	0
E 11	+203.000	+76.000	+0.000	+0.000	+0.000	+0.000	0

F1:INS/LINE F2:DEL/LINE F3:PAGE/UP F4:PAGE/DOWN F5:DATA/E {> : QUIT

Edit description :

EDIT PROGRAM NO : Input program number.（The number between 0 to 999）. Last program number in the brackets

Z : Z value is the starting coordinate value of " machining depth". It is always represented by absolute coordinate.

R : When R≠0, it's setting as "Radius & equiangular position". The X and Y are the center of the arc.

Times : When TIMES=0, the holes in the column are set to absolute coordinate. When TIMES is not 0, the holes in the column are set to increment coordinate.

F1:INS/LINE : To insert a column. **F2:DEL/LINE :** To delete a column.

F3:PAGE/UP : Page up. **F4:PAGE/DOWN :** Page down.

F5:DATA/END : "Cursor" moves to any position, then press F5 to set ending point in the column, the point is represented by " E ".

Parameter setting :

```
 EDIT PROGRAM NO.: 1     (1  )     ABS(TIMES=0),INC(TIMES=OTHER'S)

Tool Diameter          : +10.000      Tool No. (T)           :     2
Tap Pitch              :   1.25       Tool Dia.   Compen No. :     2
Tap Direction (0=+,1=-):   0          Tool Length Compen No. :     2
Safe Height            :   5.000      C Axis DTH             :   +0.000
Z Final Depth          :  10.000      C Axis STH             :   +0.000
Z Cutts                :   5.000
Clearance              :   2.000
Rotation (X,Y)         :  +0.000
         ABS X         :  +0.000
              Y        :  +0.000
Nx:      1          Ny:    1
Dx:     +0.000      Dy:  +0.000
Feedrate               : +200
Sleeping (0=Off,1=On)  :   0
Display :    0       **:   1
Index No.              :   1
Image (0=Off,1=X,2=Y)  :   0
      Distance         :  +0.000
```

＊Press the QUIT key to go back TAP screen.

Parameter description :

Tool Diameter : Set Tap diameter.

Tap pitch : Set Tap pitch.

Safe Height : When the hole position is moved, in order to prevent collide of mould, the spindle will moves to the position of "Safe Height " + " Z axis starting coordinate".

Clearance : To prevent " Over travel " caused by immediately stop.

Feedrate : To adjust feeding rate, use『F% Feedrate adjusting knob』from 10% to 150%.

Sleeping (0=Off,1=On) : 1 is to activate『sleeping switch』.

Tool No. (T) : Input tool No.

Tool Dia. Compn No. : Input diameter compensation no.

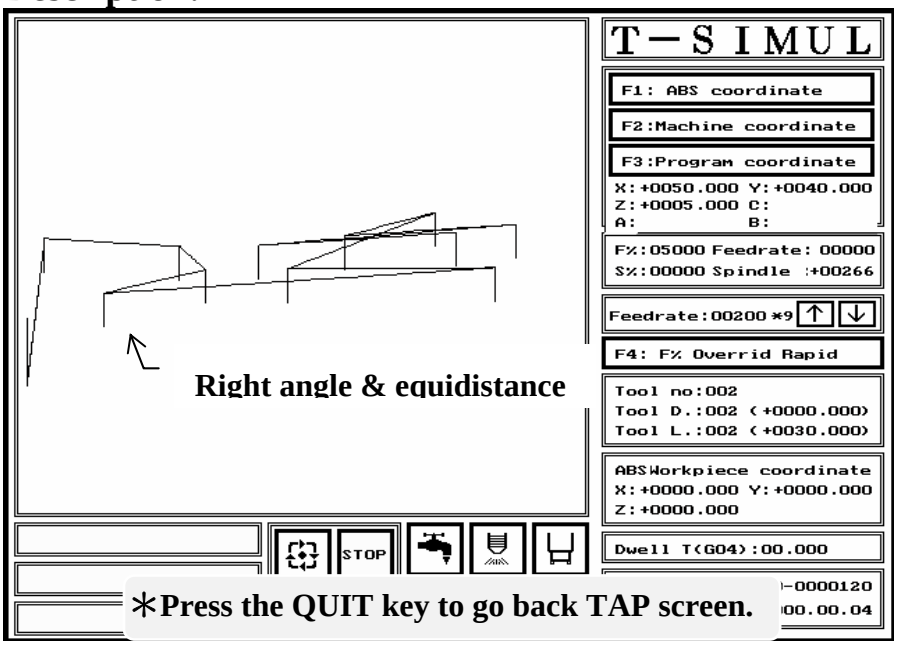
Tool length Compn No. : Input height compensation no.

C Axis Dth. : Incremental C axis angle. If non zero, the NY parameter is assumed to refer to the number of repeats in C.

C Axis Sth. : Initial C axis angle.

F2: DRAW

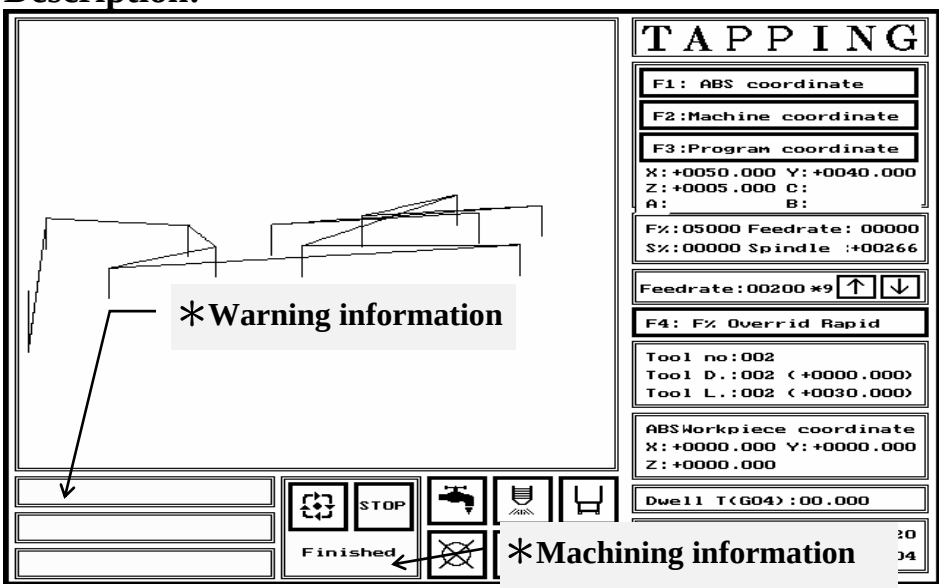
Description:



＊Press the QUIT key to go back TAP screen.

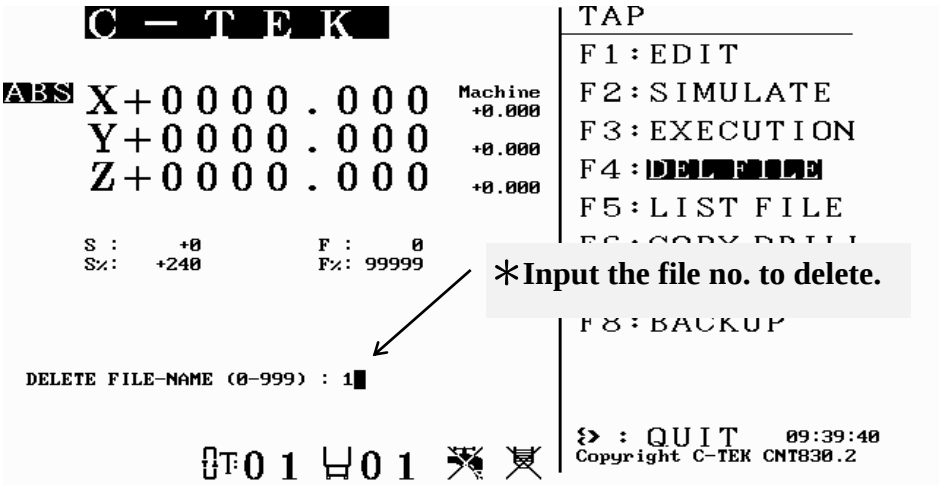
F3: EXECUTION

Description:



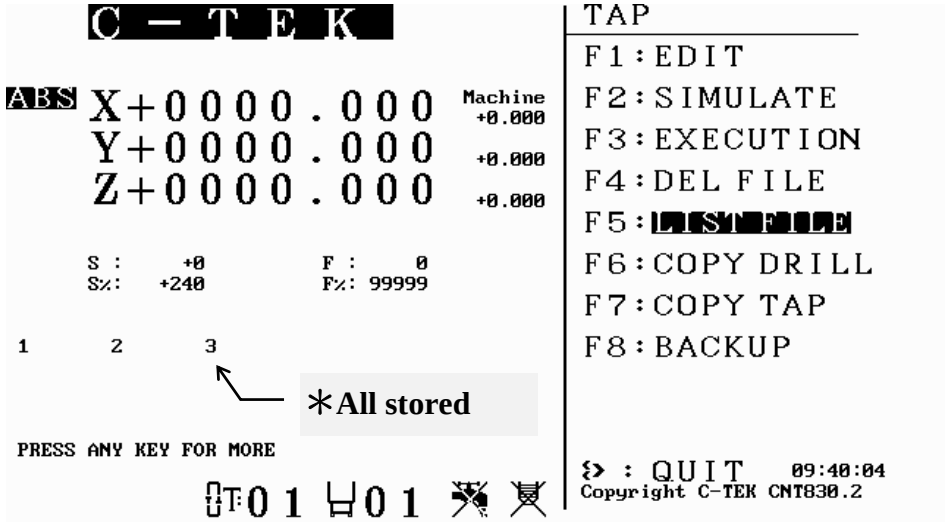
F4: DEL FILE

Description:



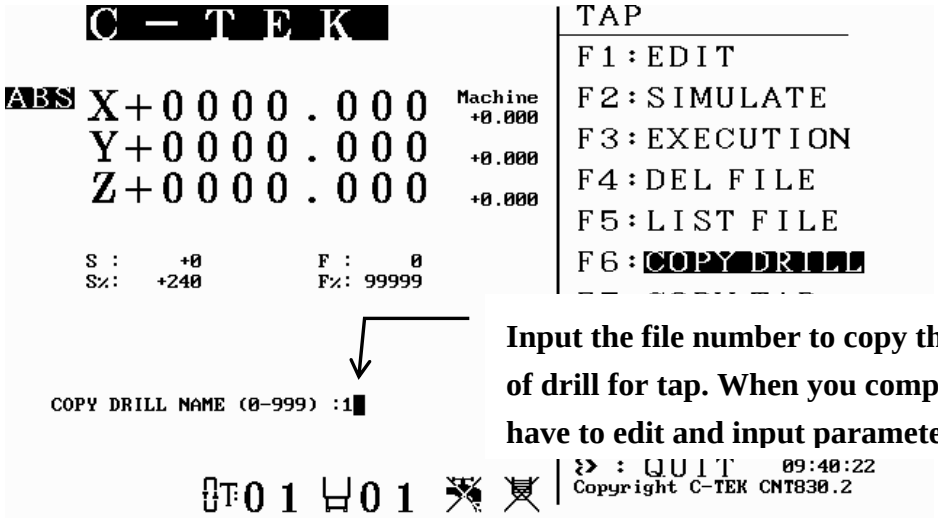
F5: LIST FILE

Description:



F6: COPY DRILL

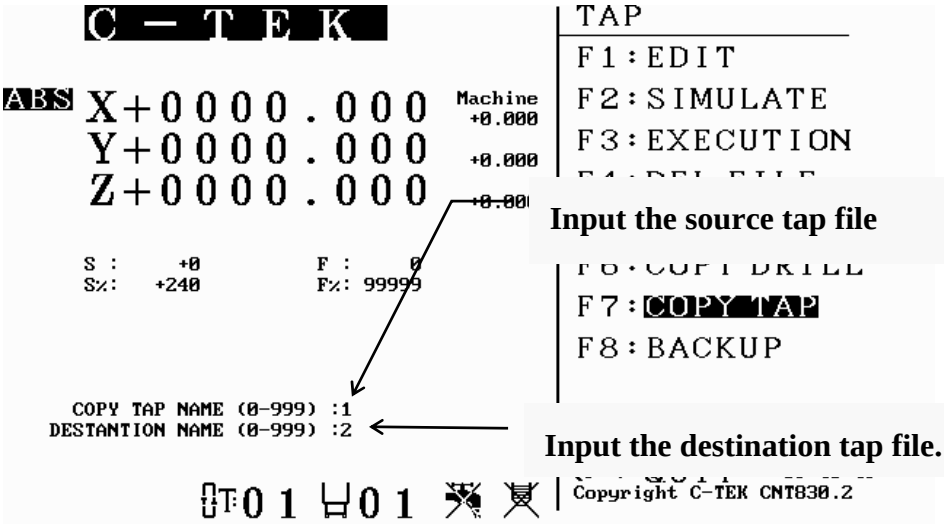
Description:



Input the file number to copy the edited program of drill for tap. When you complete copy, you have to edit and input parameter again.

F7: COPY TAP

Description:A

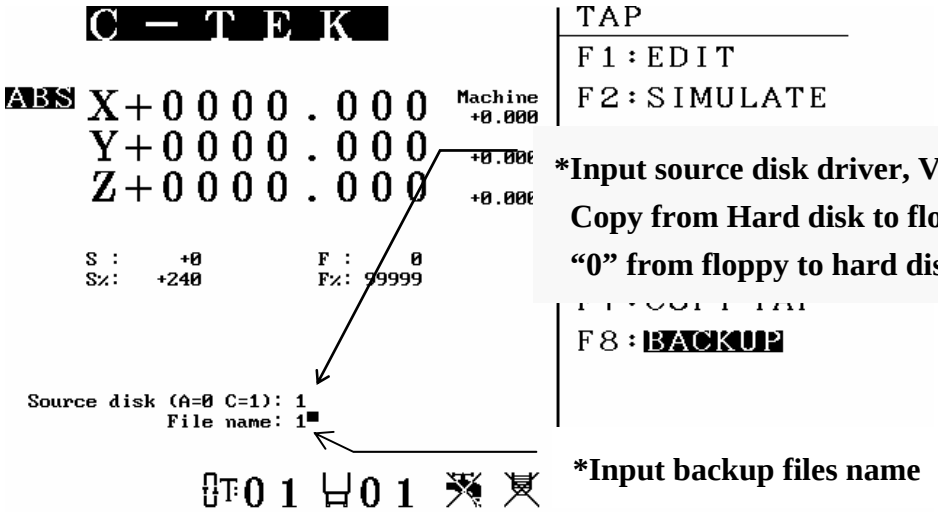


Input the source tap file

Input the destination tap file.

F8: BACKUP

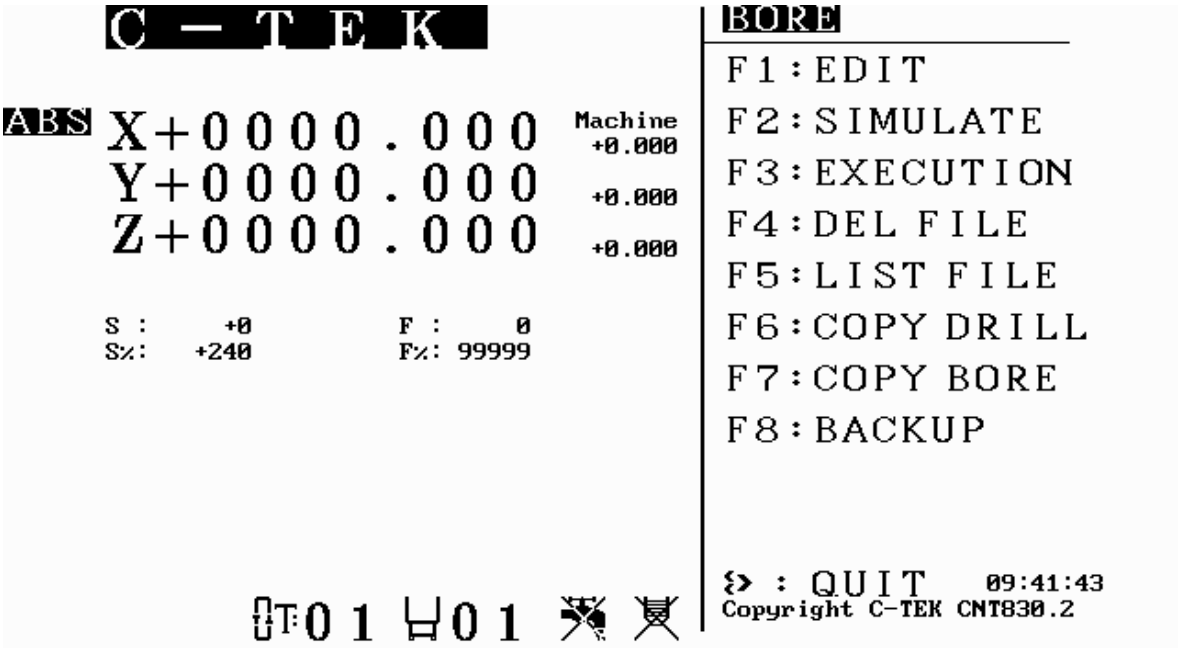
Description:



***Input source disk driver, Value "1" Copy from Hard disk to floppy disk "0" from floppy to hard disk**

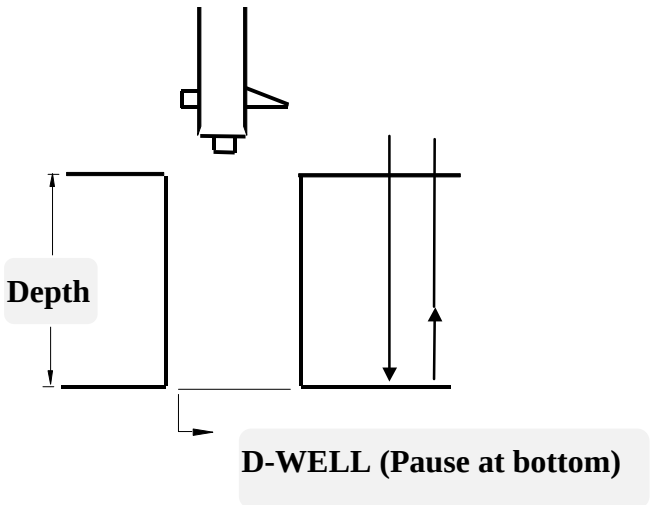
***Input backup files name**

9. BORE (Option)



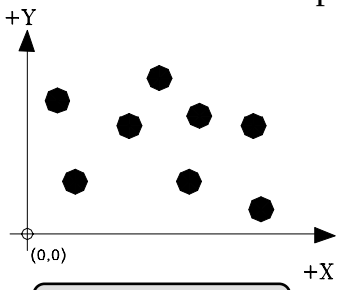
Function description:

1. Pattern of tapping: 『Normal boring』

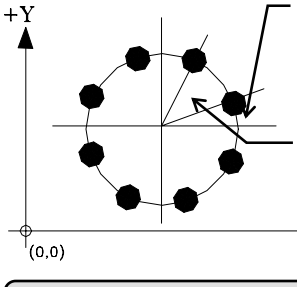


Normal boring

2. Setting of boring position: 『Random position』, 『Right angles & equidistant position』, 『Slope & equidistant position』, 『Radius & equiangular position』. These four different ways can be mixed and combined to be operated.



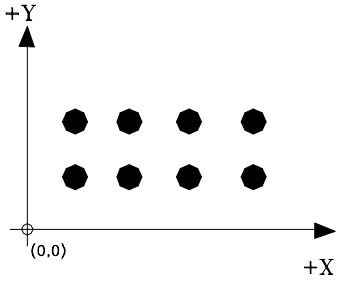
Random position



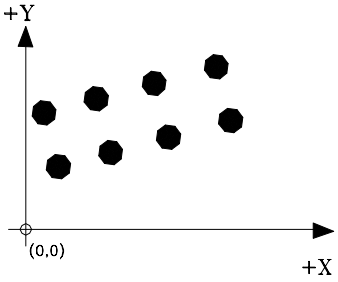
STH is the angle between the first hole and zero degree.

DTH is 『Rectangle』 degree, it means each hole is rectangle distance。

Radius & equiangular position



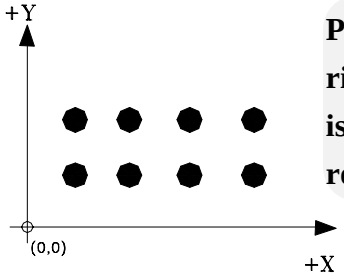
Right angle & equidistant position



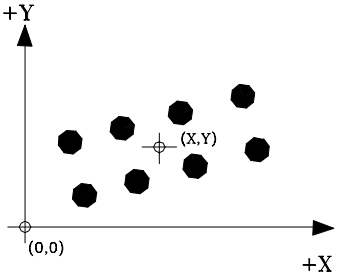
Slope & equidistant position

Caution : The 『Random position』 can be set by using of absolute coordinate.

3. Rotation X,Y(Rotating angle)**:** Angle rotation can be done at any coordinate point dedicated。



Picture shown right hand side is after angle rotating.

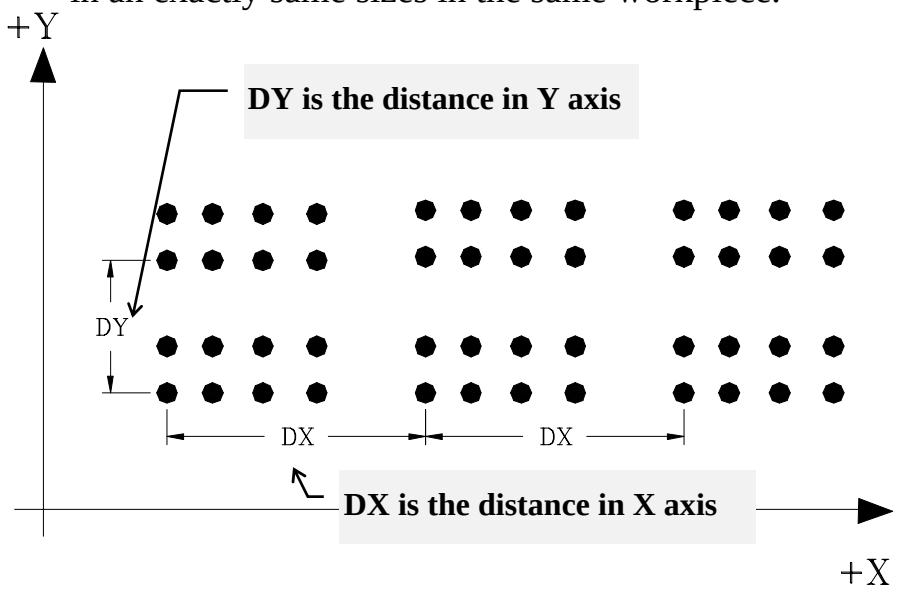


4. D-Well(Pause at bottom)**:** When boring to the desire depth, spindle can be remaining at the bottom to enable the hole bottom shining.

5. Index No.: For automatic return 『machining interrupt』 point.

Caution: For same reason you may be changed cutter when working piece was machining, you must press STOP key on remote box and record the Index no. by writing down.

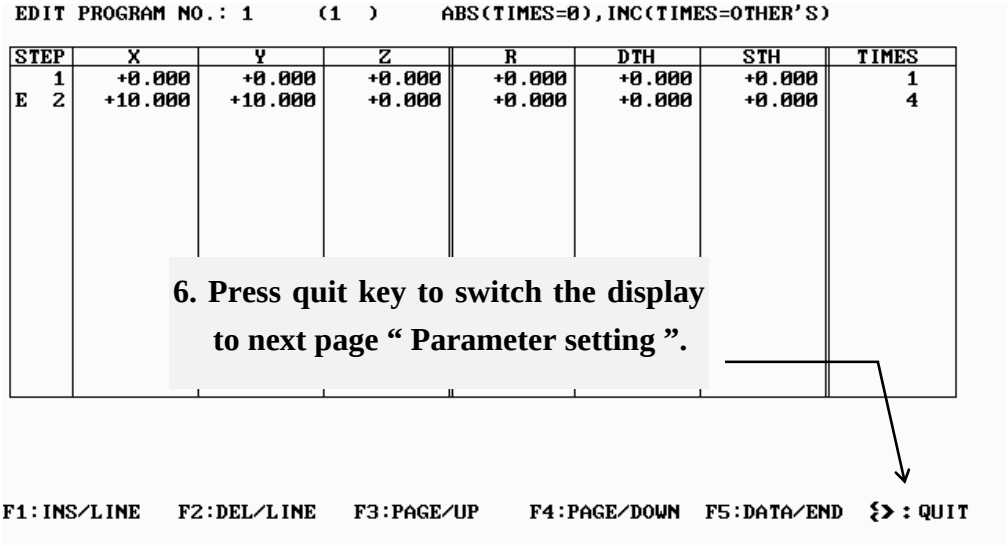
6. Nx、Ny (Receptive machining）**:** To duplicate previous machining sequence in an exactly same sizes in the same workpiece.



Caution :：The minimum setting values of Nx and Ny are 1 and 1, respectively, as picture above Nx=3 and Ny=2.

F1: EDIT

Description :


EDIT PROGRAM NO.: 1 (1) ABS(TIMES=0),INC(TIMES=OTHER'S)
STEP X Y Z R DTH STH TIMES
1 +0.000 +0.000 +0.000 +0.000 +0.000 +0.000 1
E 2 +10.000 +10.000 +0.000 +0.000 +0.000 +0.000 4
6. Press quit key to switch the display to next page " Parameter setting ".
F1:INS/LINE F2:DEL/LINE F3:PAGE/UP F4:PAGE/DOWN F5:DATA/END {>:QUIT

Edit description :

EDIT PROGRAM NO : Input program number.（The number between 0 to 999）.
Last program number in the brackets

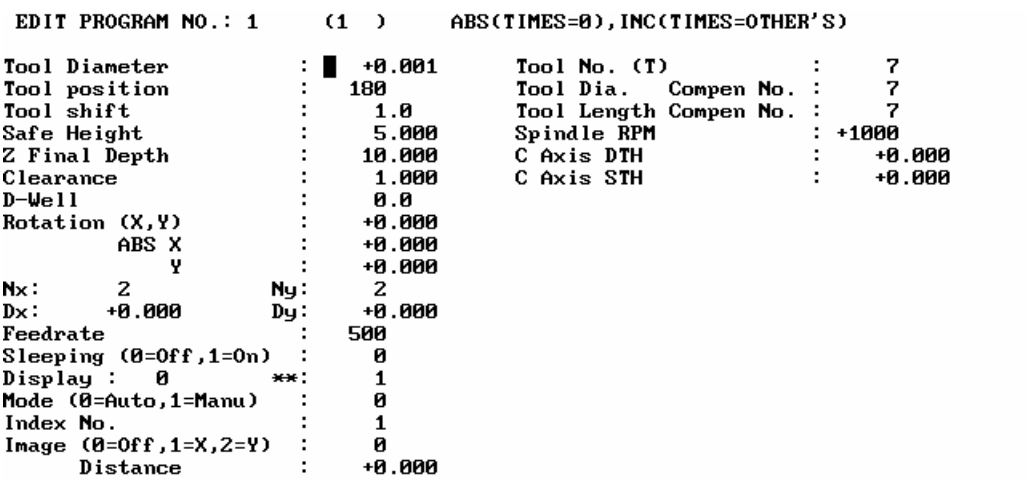
Z : Z value is the starting coordinate value of " machining depth". It is always represented by absolute coordinate.

R : When R≠0, it's setting as "Radius & equiangular position". The X and Y are the center of the arc.

Times : When TIMES=0, the holes in the column are set to absolute coordinate.
When TIMES is not 0, the holes in the column are set to increment coordinate.

F1:INS/LINE : To insert a column. **F2:DEL/LINE :** To delete a column.
F3:PAGE/UP : Page up. **F4:PAGE/DOWN :** Page down.
F5:DATA/END : "Cursor" moves to any position, then press F5 to set ending point in the column, the point is represented by " E ".

Parameter setting :


EDIT PROGRAM NO.: 1 (1) ABS(TIMES=0),INC(TIMES=OTHER'S)
Tool Diameter : +0.001
Tool position : 180
Tool shift : 1.0
Safe Height : 5.000
Z Final Depth : 10.000
Clearance : 1.000
D-Well : 0.0
Rotation (X,Y) : +0.000
ABS X : +0.000
Y : +0.000
Nx: 2 Ny: 2
Dx: +0.000 Dy: +0.000
Feedrate : 500
Sleeping (0=Off,1=On) : 0
Display : 0 **: 1
Mode (0=Auto,1=Manu) : 0
Index No. : 1
Image (0=Off,1=X,2=Y) : 0
Distance : +0.000
Tool No. (T) : 7
Tool Dia. Compen No. : 7
Tool Length Compen No. : 7
Spindle RPM : +1000
C Axis DTH : +0.000
C Axis STH : +0.000

＊Press the QUIT key to go back BORE screen.

Parameter description :

Tool position : The position (180°).is for encoder to detect where to be stopped when boring is finished. Even the tool is not stop in the same degree, encoder still can detect it's position and automatic calculate it's opposite side for tool shift (tool retract without returning mark)

Safe height : When the hole position is moved, in order to prevent collide of mould, the spindle will moves to the position of "Safe Height" + "Z axis starting coordinate.

Clearance : To prevent " Over travel " caused by immediately stop.

Feedrate : To adjust feeding rate, use『F% Feedrate adjusting knob』from 10% to 150%.

Sleeping (0=Off,1=On) : 1 is to activate『sleeping switch』.

Mode (0=Auto,1=Manu) : In manu states (1=manu), when boring cycle is completed, Z axis will be stopped at the bottom, then you will have to perform "Tool Shift" by using JOG keys or HANDWHELL(MPG) on the Remote Control Box. If you press "AUTO CYCLE" key on the Remote Control Box again, the machine will continue boring next position.

Spindle RPM : Input the spindle rpm

Tool No. (T) : Input tool No.

Tool Dia. Compn No. : Input diameter compensation no.

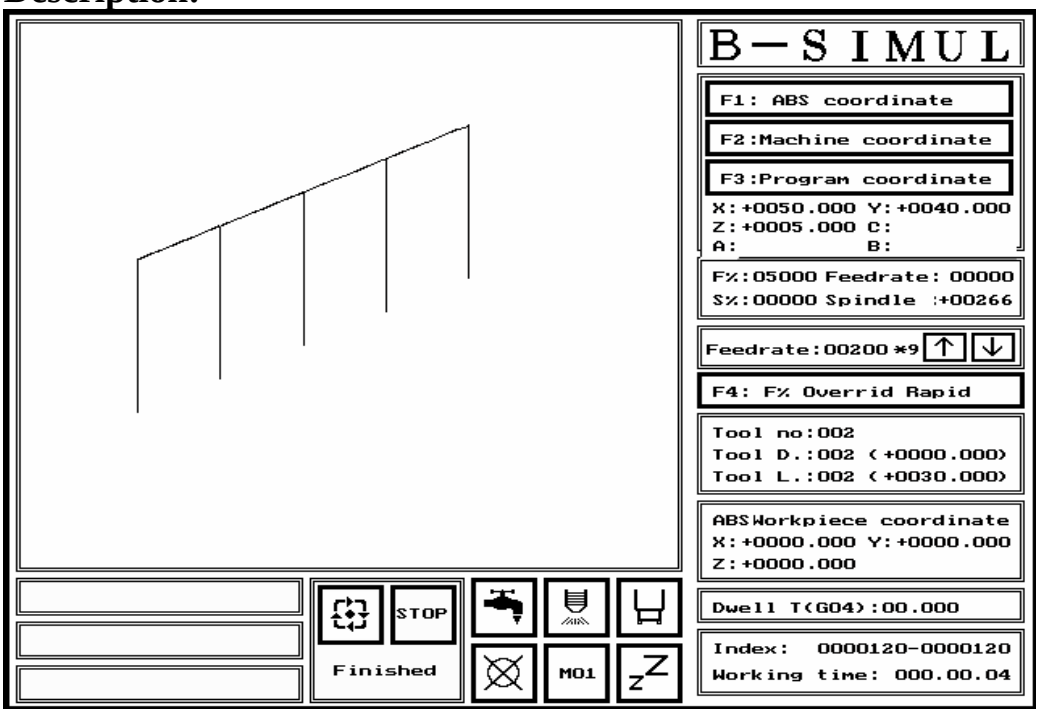
Tool length Compn No. : Input height compensation no.

Spindle RPM : Input the spindle rpm

C Axis Dth. : Incremental C axis angle. If non zero, the NY parameter is assumed to refer to the number of repeats in C.

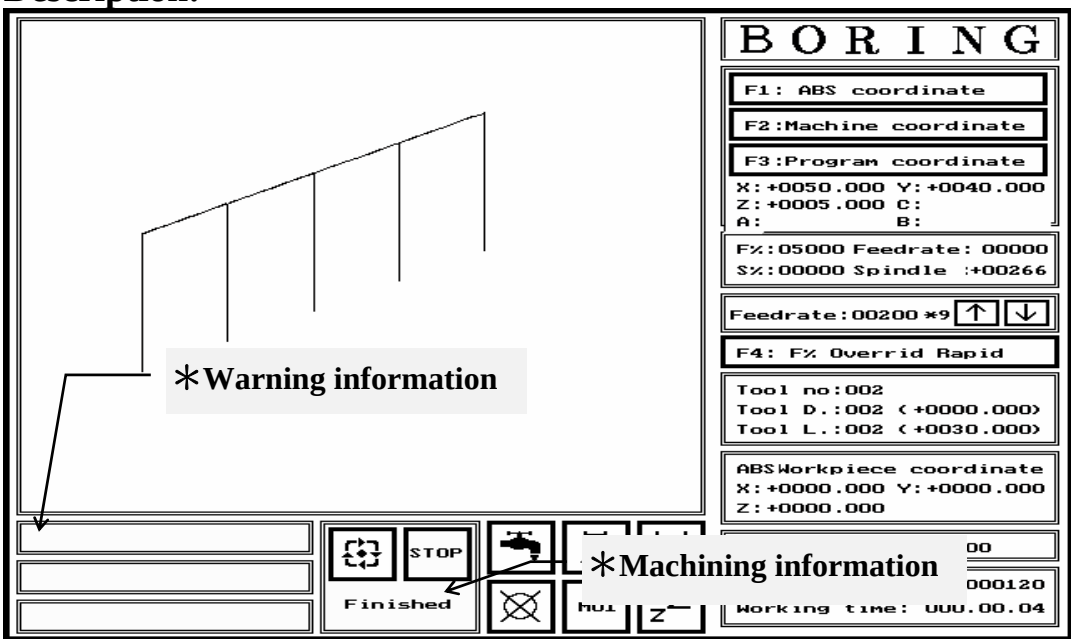
C Axis Sth. : Initial C axis angle.

F2: DRAW
Description:



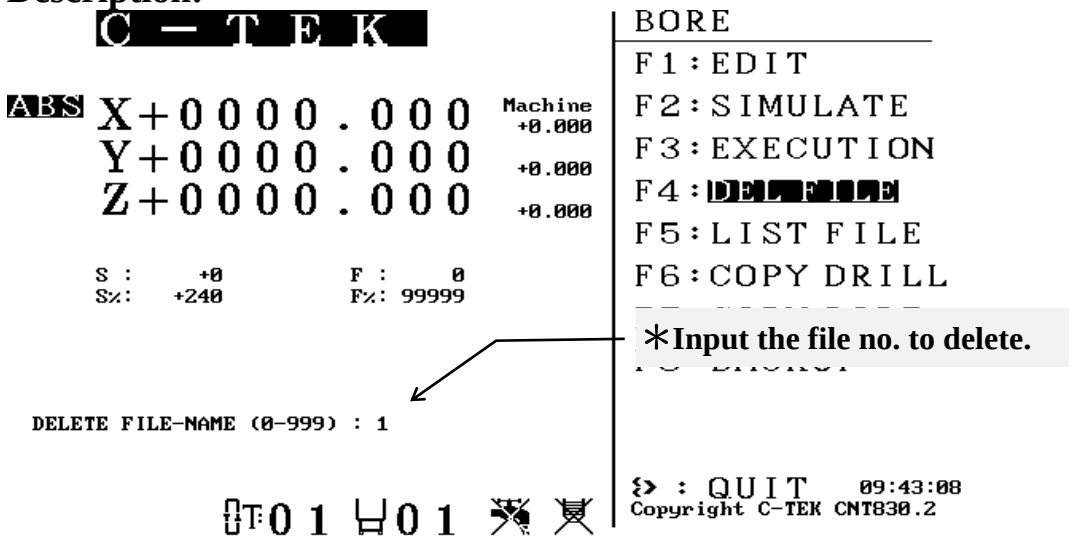
＊Press the QUIT key to go back BORE screen.

F3: EXECUTION
Description:

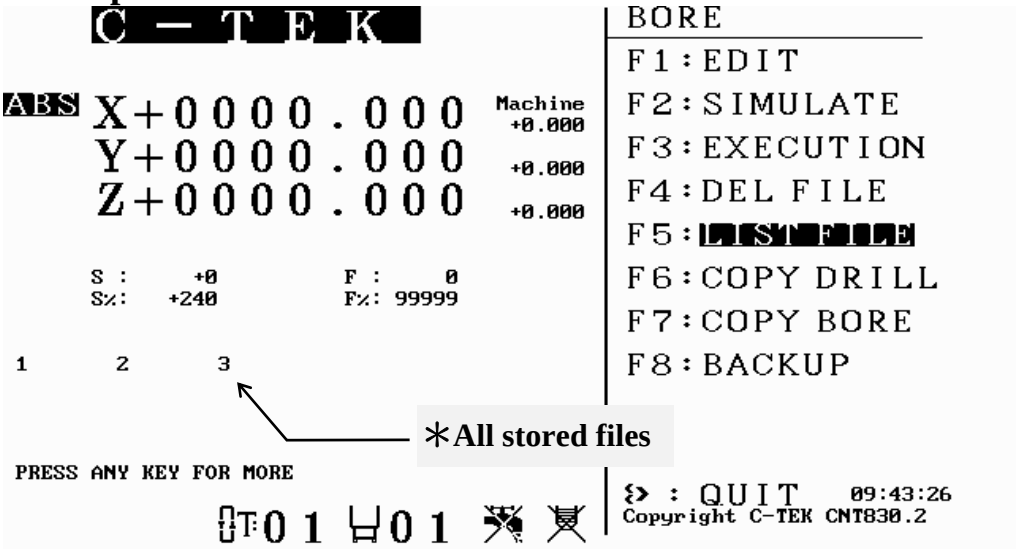


＊Press the QUIT key to go back BORE

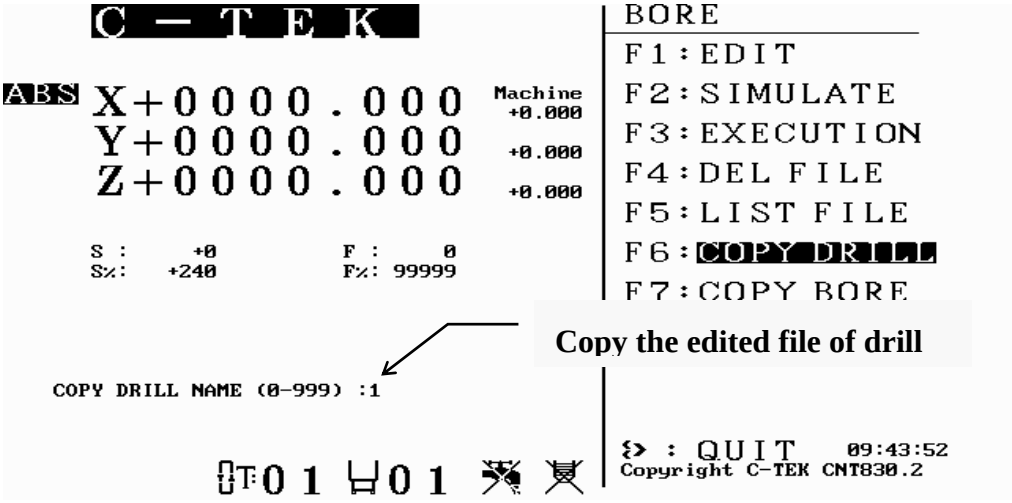
F4: DEL FILE
Description:



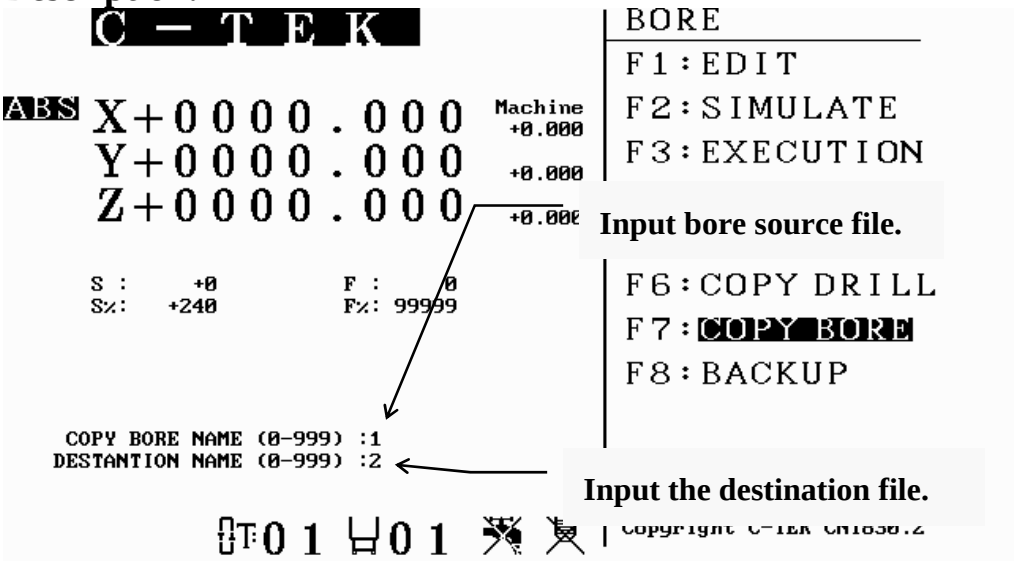
F5: LIST FILE
Description:



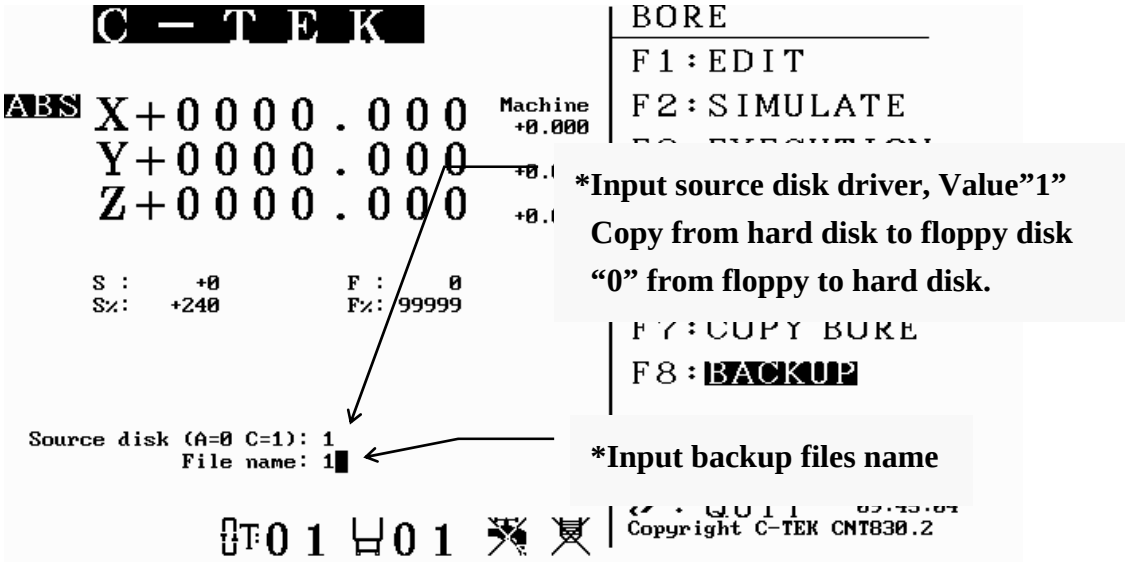
F6: COPY DRILL
Description:



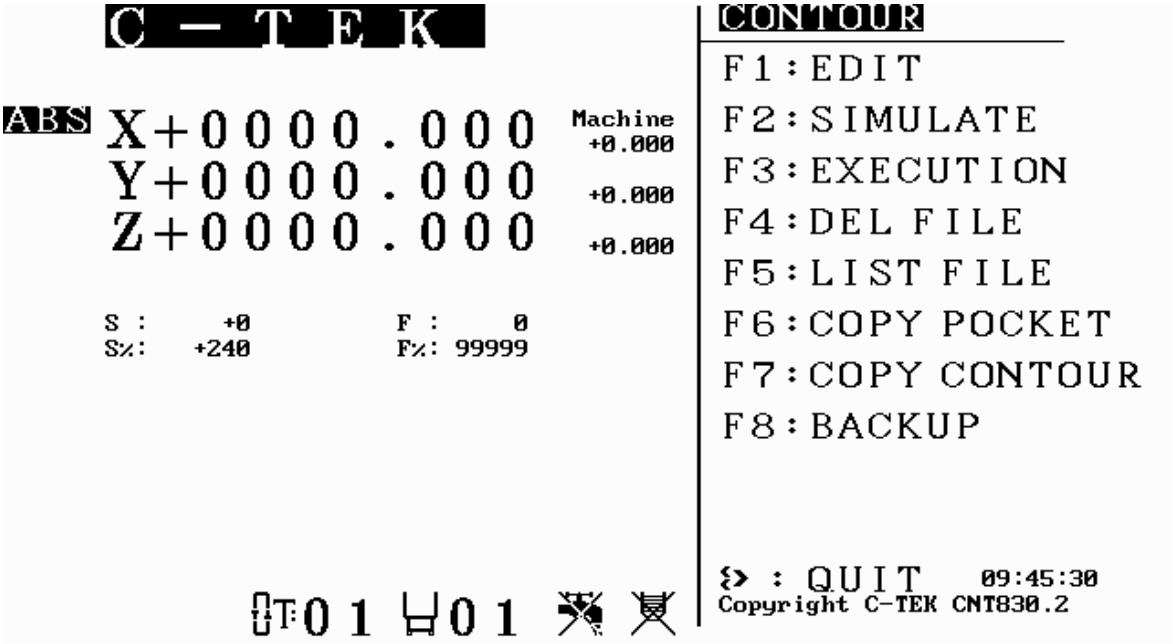
F7:COPY BORE
Description:



F8: BACK UP
Description:

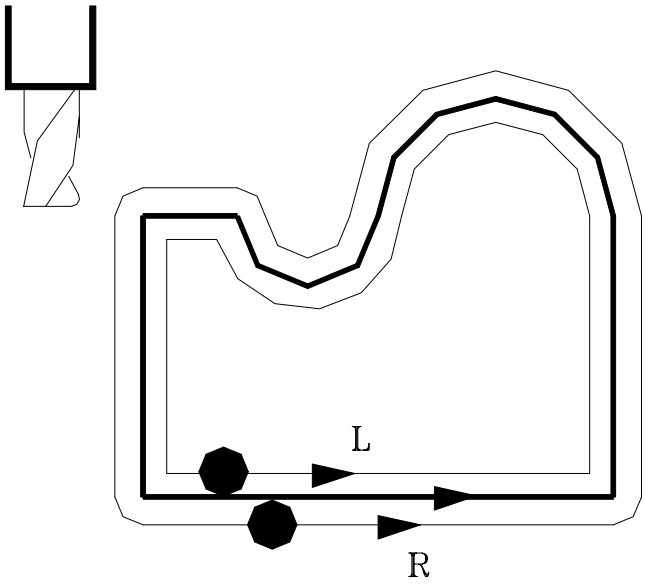


10. CONTOUR

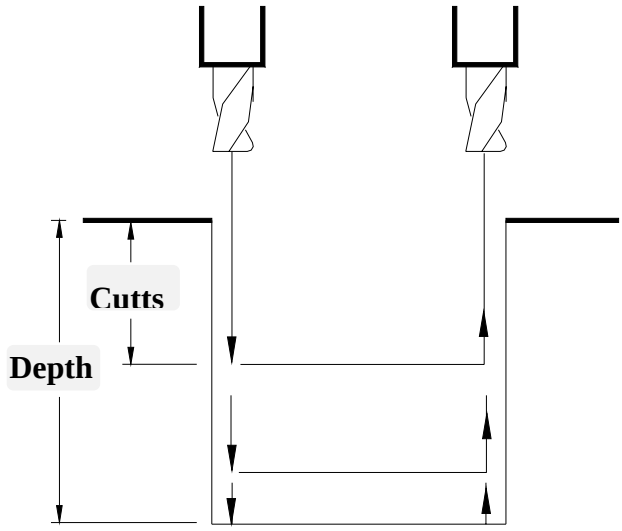


Function Specification:

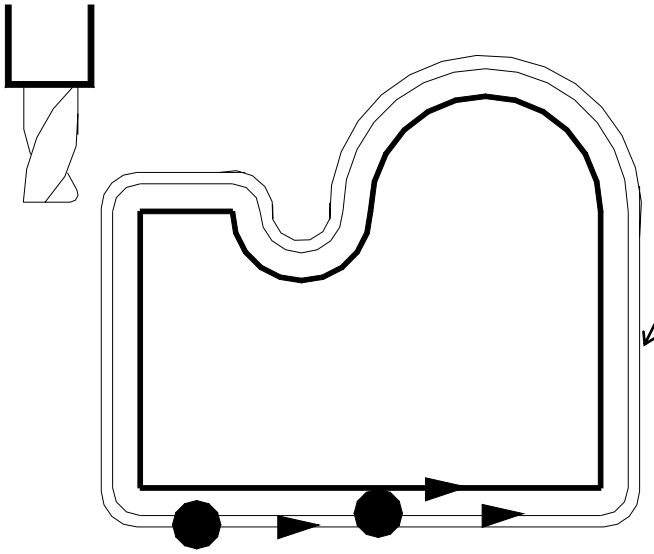
1. Tool Diameter Compensation:『R Right Offset』, 『L Left Offset』.



2. Cycle Milling:『Machining Depth』cycle milling can be sequentially controlled.



3. Cut Mode (Automatic Rough/ Finish machining)：『rough+finish』,『rough』,『finish』.



＊In between, the space is referred as『Finish machining amount』.

＊『Finish Machining』：

1. Setting point and Starting point of the tool come from the side direction. The space in between is referred as the『Finish machining amount』.

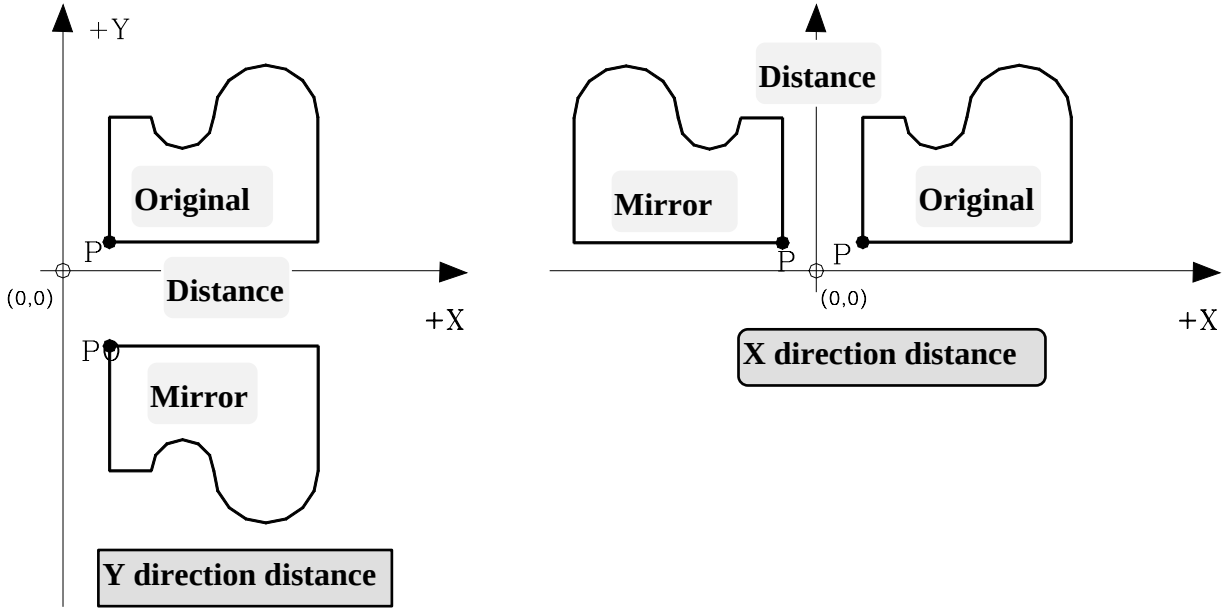
2. Machining path is the same the indicated direction.

3. To improve the polish on the surface, user can set Reverse Milling.

＊『Rough』: The machining process can be selectively controlled and its path direction is the same as the indicated one on the figure.

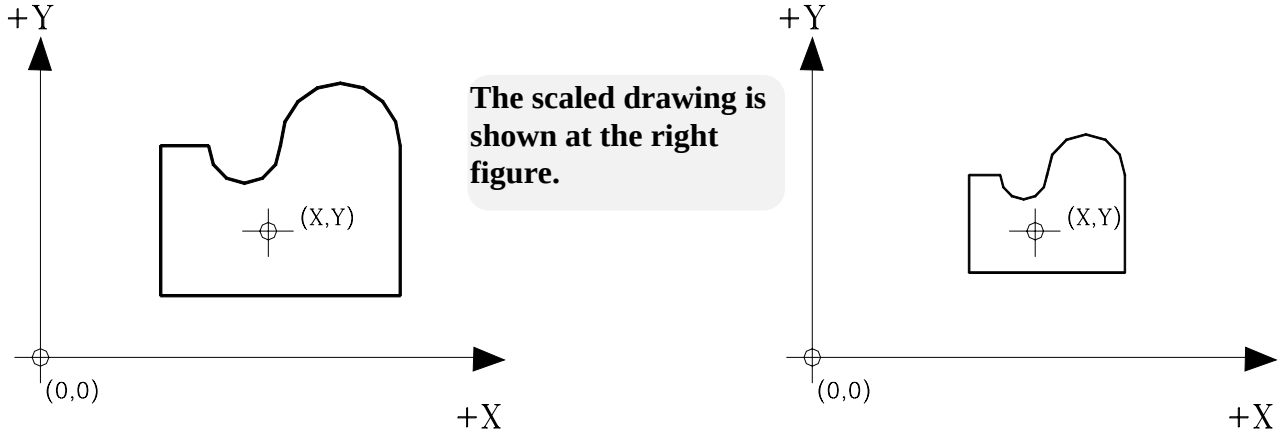
4. Reverse (Reverse Milling)**:** For finish machining, you can reverse the machining path to polish the surface of the workpiece.

5. Image(Mirror Machining)**:** User can do the mirror projection of the『X direction distance』and『Y direction distance』and『XY direction distance』.

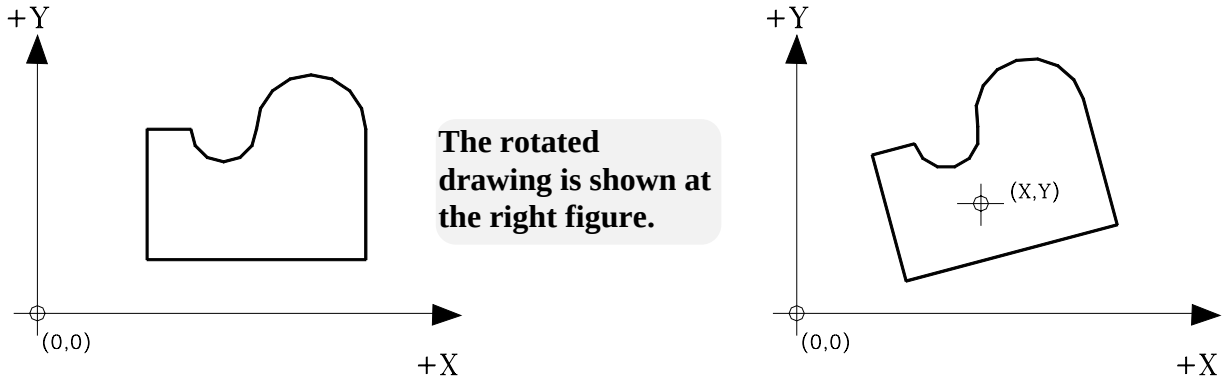


Note: The mirror projection is based on the 『P point 』of the tool.

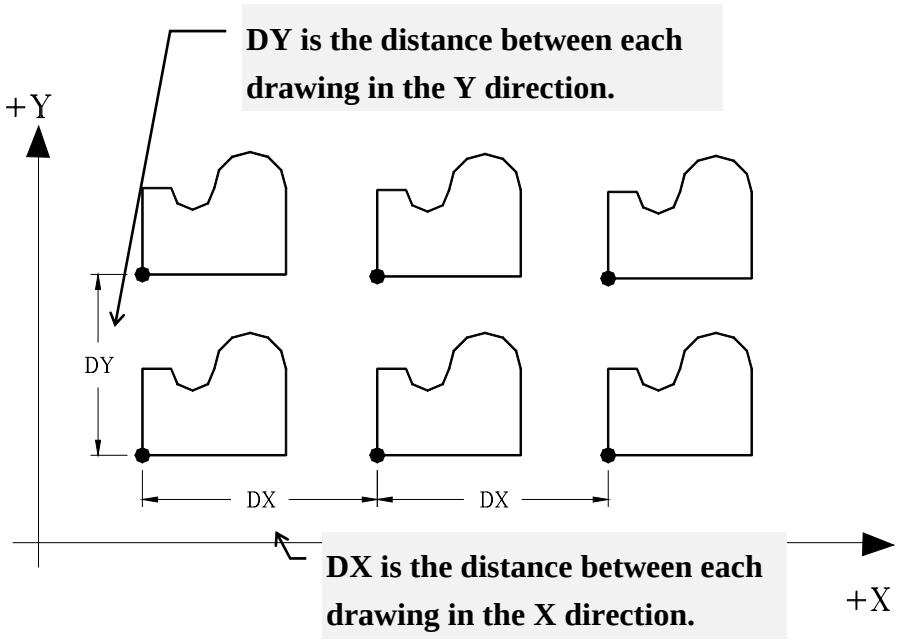
6. Scale: User can scale the existing entity. The scale factor ranges from 0.001 to 9.999.



7. Rotation (Rotating angle): User can rotate the existing entity.



8. Nx、Ny(Repetitive Machining): User can input one machining drawing to perform multiple machining for the same object by specifying Nx and Ny.



Note：The minimum values of NX and NY were set to 1 and 1, respectively. In the above figure, NX=3 and NY=2.

9. Cut way (0=No, 1=Yes) : to determine how the tool will descend into the metal.

Option	Description
0	If Cut Way is zero (Off), it will simply descend the tool at the start of the (offset) contour, at the Z Feedrate, to the current Z depth. At the end of the contour the tool is retracted to the Safe height.
1	If there is a position move prior to the start of the contour, It will use this point to descend the tool, then approach the contour using a linear move followed by an arc which is tangential to the start of the contour. At the end of the Contour, It adopts the reverse strategy and returns to the start point, using a tangential arc to detach from the contour. Once back at the start point the tool is retracted. If there is no position move before the start of the contour, It will descend the tool (at the Z feedrate) until it just touches the metal, and then it will descend the tool to the current Z depth as it machines to a point 50% along the first span of the contour. At the end of the contour, It will finish the machining of the first span before retracting.

10. Taper Degree : An angle measured from the vertical, it causes offsetting to be increased as the cutting depth increases in order to produce contours with tapered sides.

11. Taper Finish Option : Finish machining only occurs if the Finish allowance is non zero and the Cut Mode is 1 or 3. If Taper Degree is zero, then this parameter is ignored.

Option	Description
1	The finish machining of the contour takes place at constant Z level, after rough machining at each Z depth, then finish machining.
2	The finish machining of the contour consists of 3 axis XYZ moves. All cutting moves are in the Z direction, at the bottom of each move the tool retracts at rapid to the ***Clearance*** plane and moves in XY to the start of the next move. It then descends to Z zero, and then in three axes to depth. The distance (in the XY plane) between successive cutting passes is controlled by the ***Taper Surface Tolerance*** parameter.
3	Similar to option 2, but the tool cuts in both directions, i.e. It cuts down in three axes to ***Z Final Depth***, moves in XY to the bottom point of the next cutting pass, then move upwards to the Top surface. This produces a much shorted cycle time than option 2, but may not be suitable in all materials.

12. Taper Surface Tolerance : This parameter is used to control the surface finish achieved when the Taper Finish Option is set to either 2 or 3. The distance between successive passes is given by

d = Square root((D * A) – (A * A))
where d = distance, D=ToolDiameter and A is the Surface Tolerance.

F1:Edit Description :

Input program number 45.（The number between 0 to 999）.

＊ Last program number in the brackets.

```
EDIT PROGRAM NO.: 1      (1   )  Coordinate System : 0 (ABS=0,INC=1)

Step | Cmd | Fn |      X      |      Y      |      Z
   1 |  P  |  0 |   +50.000   |    +0.000   |    +0.0
   2 |  L  |  0 |  +100.000   |    +0.000   |    +0.0
   3 |  L  |  0 |  +100.000   |   +75.000   |    +0.0
   4 | CCW |  0 |   +50.000   |   +75.000   |    +0.0
   5 | CW  |  1 |   +20.000   |   +75.000   |    +0.0
   6 |  L  |  0 |    +0.000   |   +75.000   |    +0.0
   7 |  L  |  0 |    +0.000   |    +0.000   |    +0.0
   8 |  L  |  0 |   +50.000   |    +0.000   |    +0.000    +0.000    +0.000    +0.000
```

0 means the drawing is based on the Absolute coordinates. 1 means the drawing is based on the incremental coordinate.

＊Z value is the starting coordinate value of " machining depth", it is always represented by absolute coordinate.

Fn is a help function code for intersection or tangency.

```
F1:INS   F2:DEL   F3:PG/UP   F4:PG/DN   F5:DATA/END   F6:CHK/DRAW   {> : QUIT
```

Note : Function key define

F1:INS : Insert a column **F2:DEL :** Delete a column

F3:PG/UP : Page up **F4:PG/DN :** Page down

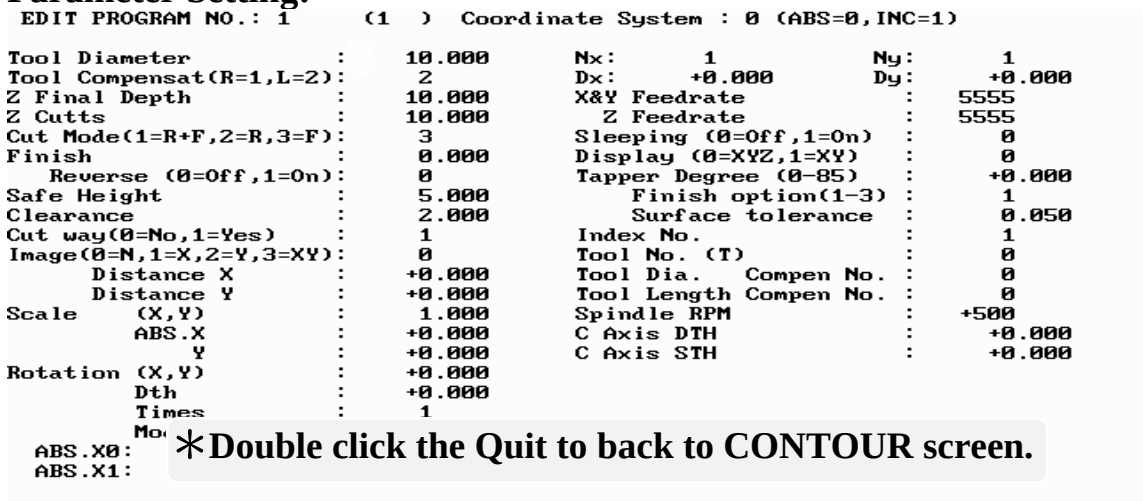
F5:DATA/END : Set end column

F6:CHK/DRAW : Cursor moves to any position, then press F6 to the pocketing graph inspection screen. Check whether this is correct and machining is executed at this program line.

QUIT : to switch the display to next page " Parameter setting".

＊ CAUTION：If the command "P" put in both STEP1 and STEP2, then the first P indicate the cutting point and the 2nd P indicate the cutting point of programming.

Parameter Setting:



Parameter description:

Tool diameter : The diameter of the tool to be used, must be greater than zero.

Z final depth : The final depth of the contour

Z Cutts : The incremental z depth. If Z Cutts is less than Z final depth, it produces a number of passes in Z, at increments of Z Cutts in order to achieve the final depth. The final pass may be less than Z Cutts.

Cut Mode : 1 = Roughing + Finishing, 2 = Roughing only, 3 = Finishing only

Finish : Finish allowance for final machining. The Finish machining will only take place if this parameter is non zero, and Cut Mode specifies either 1 or 3

Reverse : 1 = Reverse direction of finish cut, 0 = finish cut is in the same direction as the geometry.

Safe Height : The assumed Z value of the tool at the start of the program, Z position for retract at end of program

Clearance : The Z value at which Feedrate changes from Rapid to Feed when the tool is descending.

Image : Mirror image flags, 0 = None, 1= Mirror about X, 2= Mirror about Y, 3= Mirror about X and Y.

Distance X Distance Y

Incremental distance from Geometry origin to mirror axis

Rotation (X,Y) : Rotation angle in the XY plane.

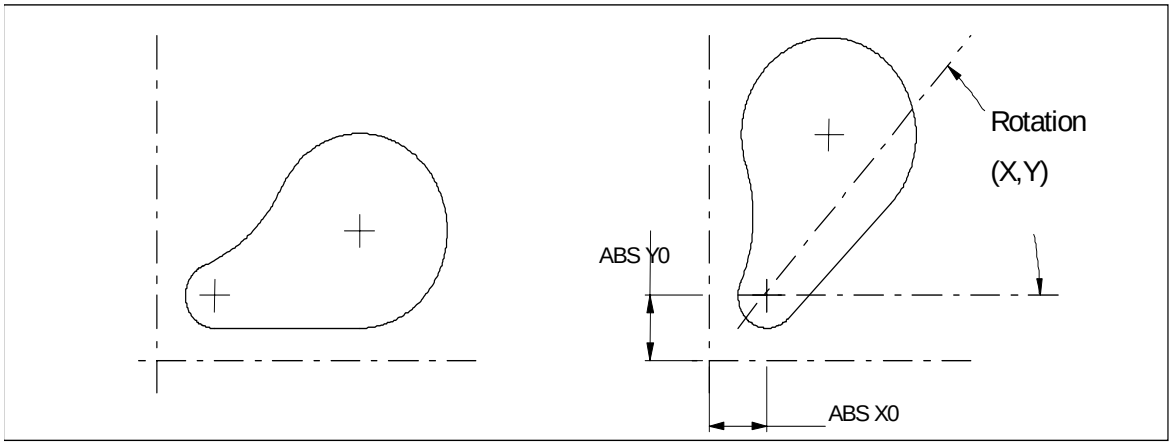
Rotation Dth : Rotation incremental angle positive counter clockwise.

Rotation Times : Number of Copies

Rotation Mode : If 1 the geometry is rotated by the angle specified above. If 0 then the geometry is translated to a new position according to the angle and radius from the rotation center.

ABS.X0 ABS.Y0

These two parameters define the XY centre of rotation used by the Rotation(X, Y) angle.

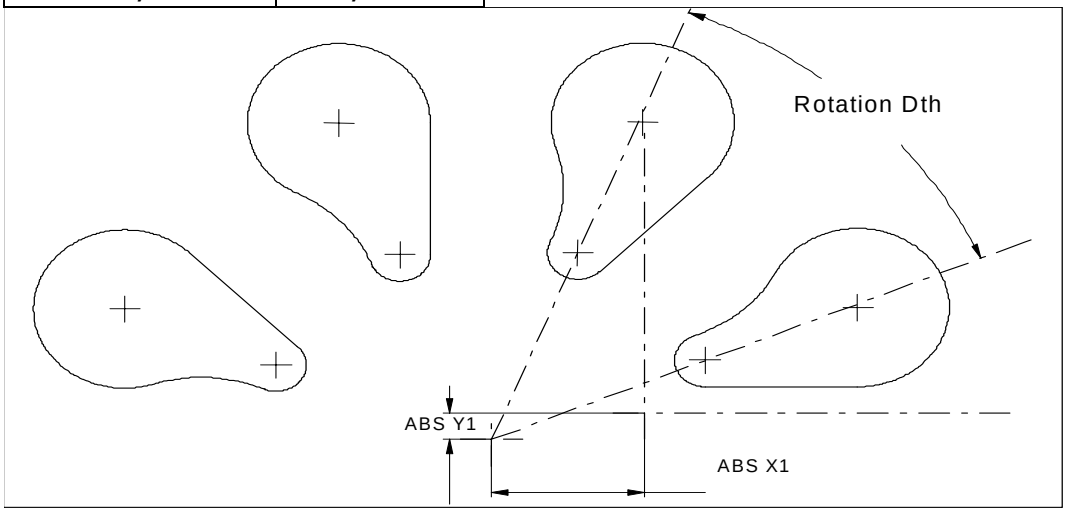


ABS.X1 ABS.Y1

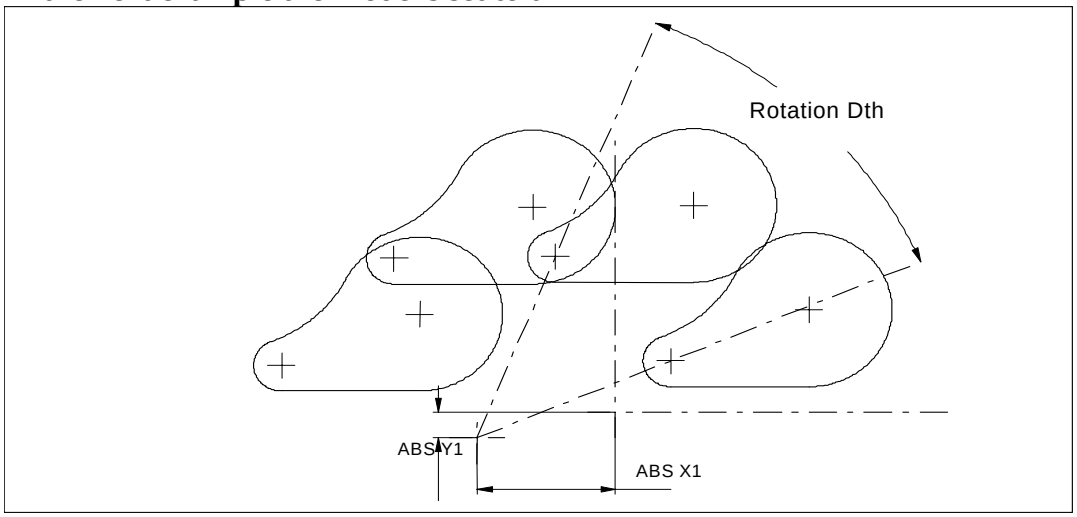
These two parameters define the point about which the geometry is rotated when producing multiple copies around an arc.

In this example the parameters are …

Rotation Dth	**45**
Times	**4**
Mode	**1**
ABS X1, Y1	**-40, -10**



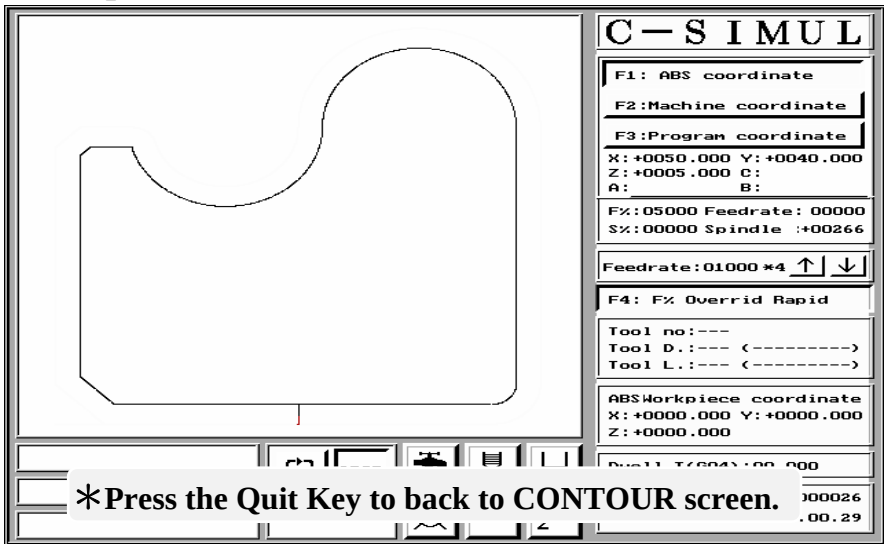
In the next example the Mode is set to 0



XY Feedrate : FeedRate for all XY (and XYZ) moves
Z Feedrate : Feedrate used when plunging into the workpiece.
Tool No. (T) : Tool number used in tool change block.
Tool Dia. Compen No.: Tool diameter compensation numbers.
Tool Length Compen No.: Tool length compensation numbers.
Spindle RPM : Spindle speed
C Axis Dth. : Incremental C axis angle. If non zero, the NY parameter is assumed to refer to the number of repeats in C.
C Axis Sth. : Initial C axis angle.

F2: DRAW

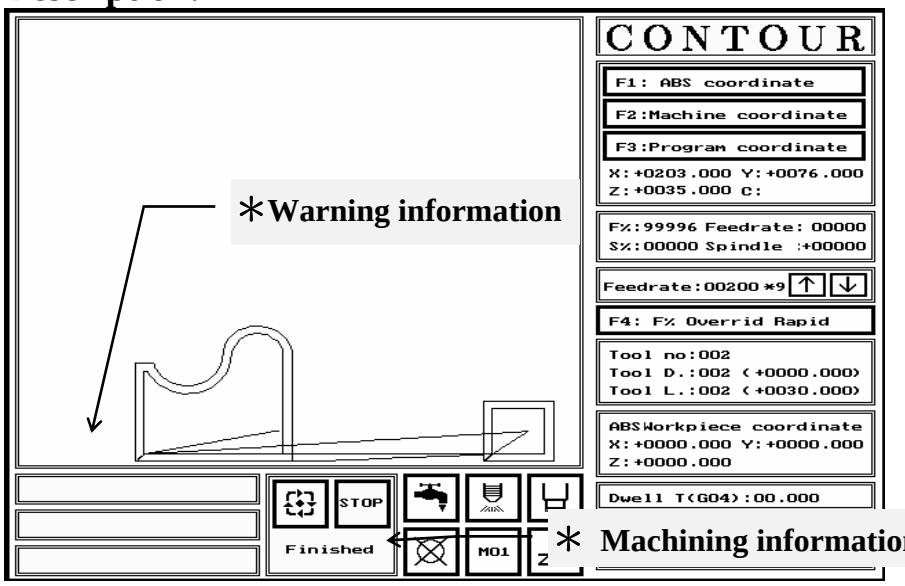
Description：



＊Press the Quit Key to back to CONTOUR screen.

F3: EXECUTION

Description:



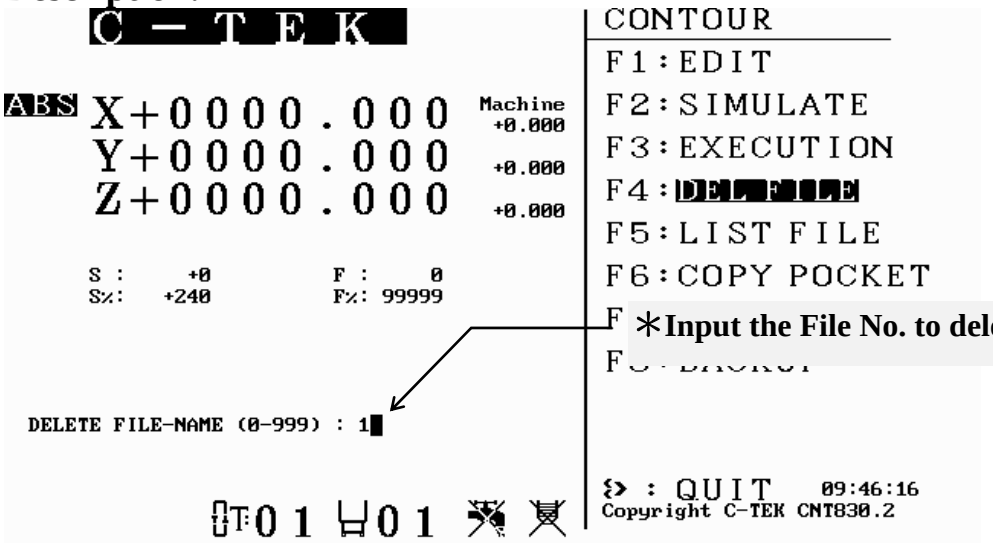
＊Warning information

＊ Machining information

＊Press the Quit key to back to CONTOUR screen.

F4: DEL FILE

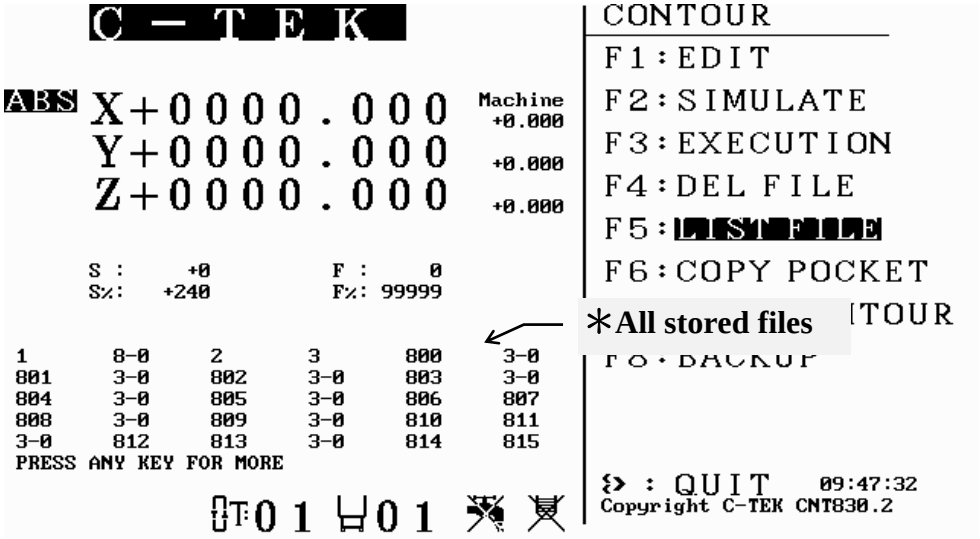
Description:



＊**Input the File No. to deletion.**

F5: LIST FILE

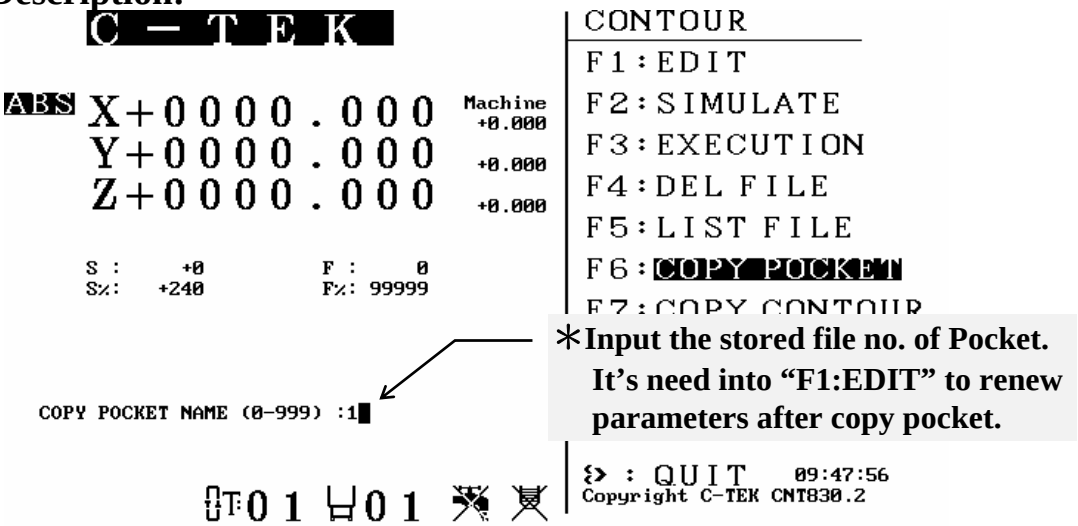
Description:



＊**All stored files**

F6:COPY POCKET

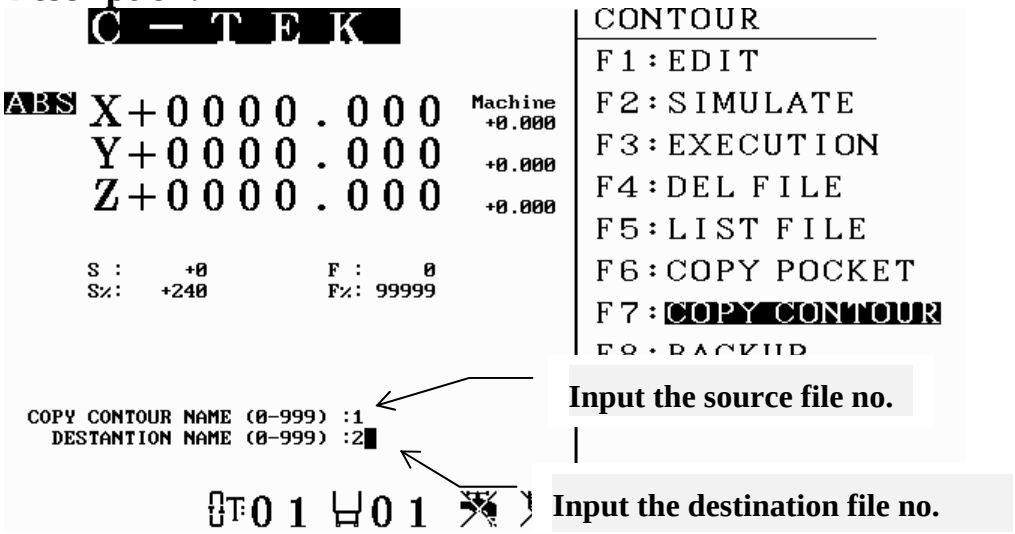
Description:



＊**Input the stored file no. of Pocket. It's need into "F1:EDIT" to renew parameters after copy pocket.**

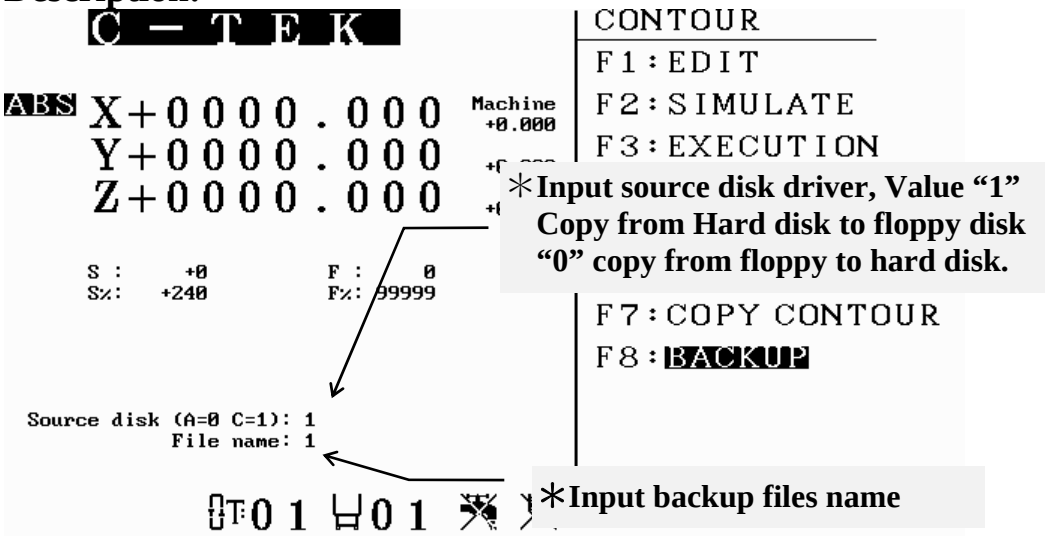
F7: COPY CONTOUR

Description:



F8: BACKUP

Description:



Sample edit program with cutting point

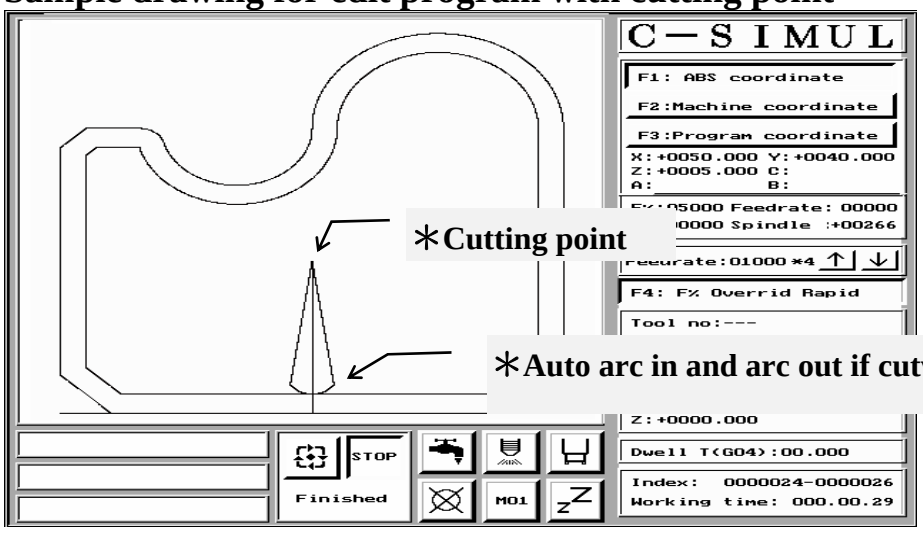
EDIT PROGRAM NO.: 1 (1) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	R+/C-
1	P	0	+50.000	+40.000	+0.000	+0.000	+0.000	+0.000
2	P	0	+50.000	+0.000	+0.000	+0.000	+0.000	+0.000
3	L	0	+100.000	+0.000	+0.000	+0.000	+0.000	+10.000
4	L	0	+100.000	+75.000	+0.000	+0.000	+0.000	+0.000
5	CCW	0	+50.000	+75.000	+0.000	+0.000	+0.000	+25.000
6	CW	1	+20.000	+75.000	+0.000	-15.000	+0.000	+5.000
7	L	0	+0.000	+75.000	+0.000	+0.000	+0.000	-5.000
8	L	0	+0.000	+0.000	+0.000	+0.000	+0.000	-10.000
9	L	0	+50.000	+0.000	+0.000	+0.000	+0.000	+0.000

＊First P means cutting point

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {> : QUIT

Sample drawing for edit program with cutting point



Sample edit parameter with taper

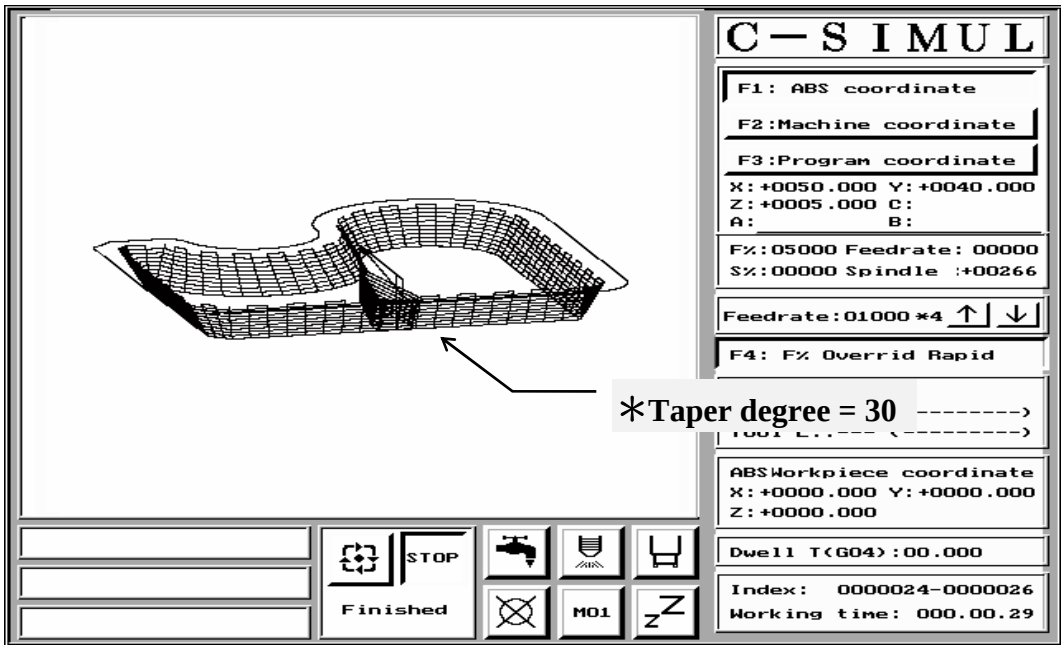
EDIT PROGRAM NO.: 1 (1) Coordinate System : 0 (ABS=0,INC=1)

Tool Diameter	:	10.000	Nx:	1	Ny: 1
Tool Compensat(R=1,L=2)	:	2	Dx:	+0.000	Dy: +0.000
Z Final Depth	:	10.000	X&Y Feedrate		: 1000
Z Cutts	:	1.000	Z Feedrate		: 500
Cut Mode(1=R+F,2=R,3=F)	:	1	Sleeping (0=Off,1=On)		: 0
Finish	:	0.000	Display (0=XYZ,1=XY)		: 0
Reverse (0=Off,1=On)	:	0	Tapper Degree (0-45)		: +30.000
Safe Height	:	5.000	Finish option(1-3)		: 3
Clearance	:	2.000	Surface tolerance		: 0.500
Cut way(0=No,1=Yes)	:	1	Index No.		: 1
Image(0=N,1=X,2=Y,3=XY)	:	0	Tool No. (T)		: 0
Distance X	:	+0.000	Tool Dia. Compen No.		: 0
Distance Y	:	+0.000	Tool Length Compen No.		: 0
Scale (X,Y)	:	1.000	Spindle RPM		: 0
ABS.X	:	+0.000	C Axis DTH		: 0.000
Y	:	+0.000	C Axis STH		: +0.000
Rotation (X,Y)	:	+0.000			
Dth	:	+0.000			
Times	:	1			
Mode	:	0			

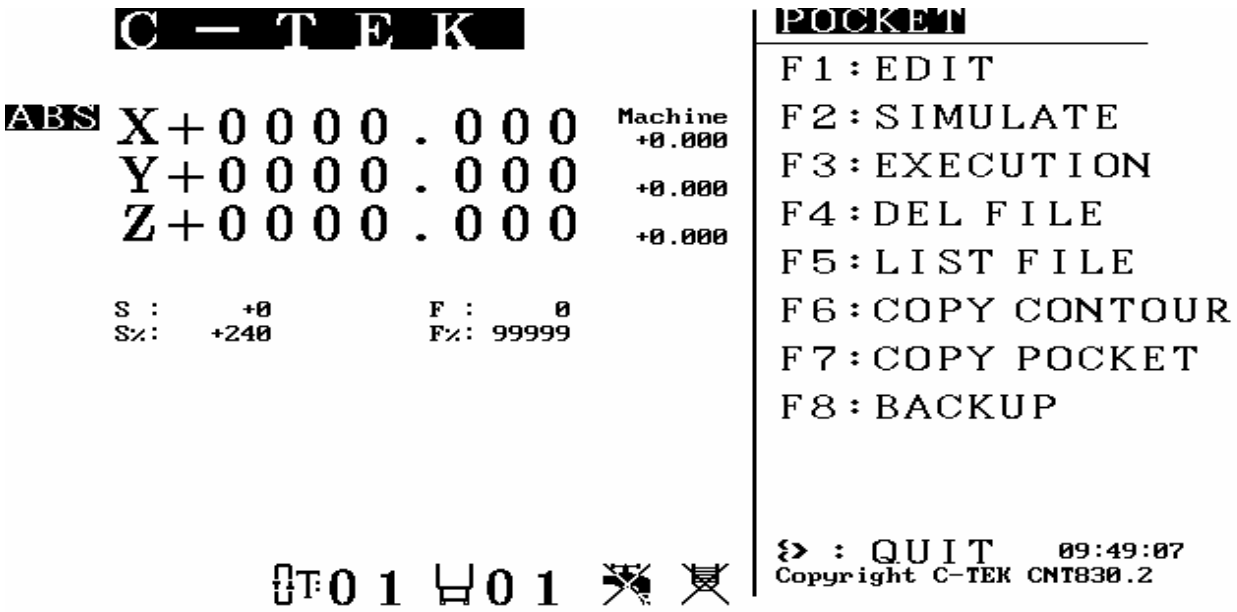
ABS.X : +0.000 Y : +0.000

＊Taper degree = 30

Sample drawing for edit parameter with taper

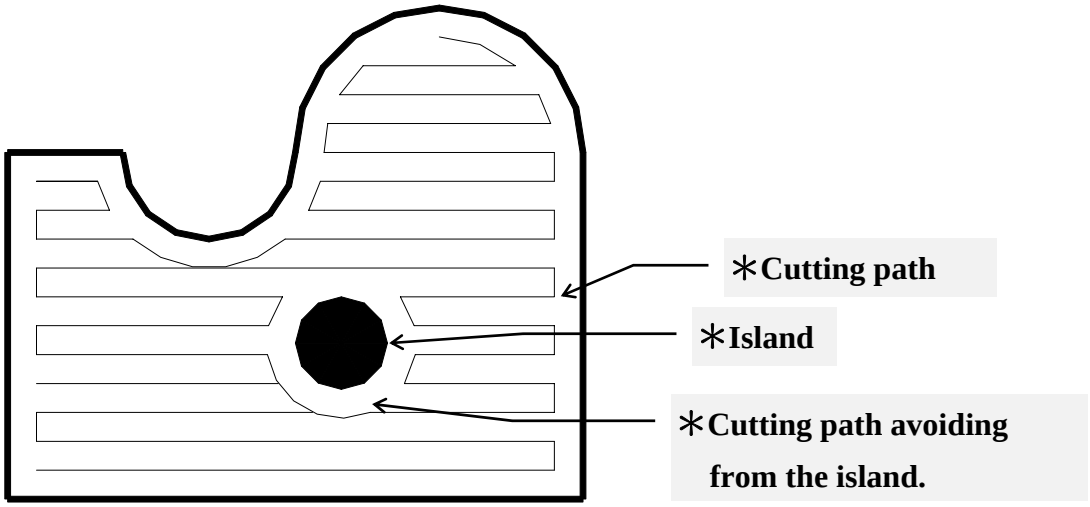


11. POCKET

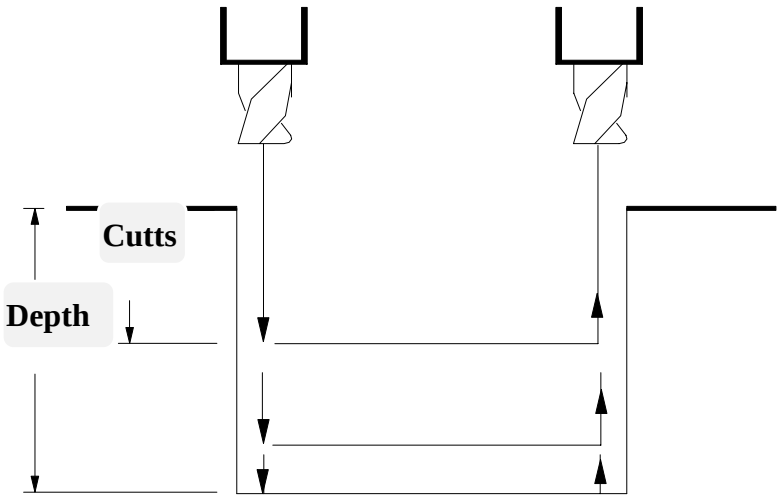


Function:

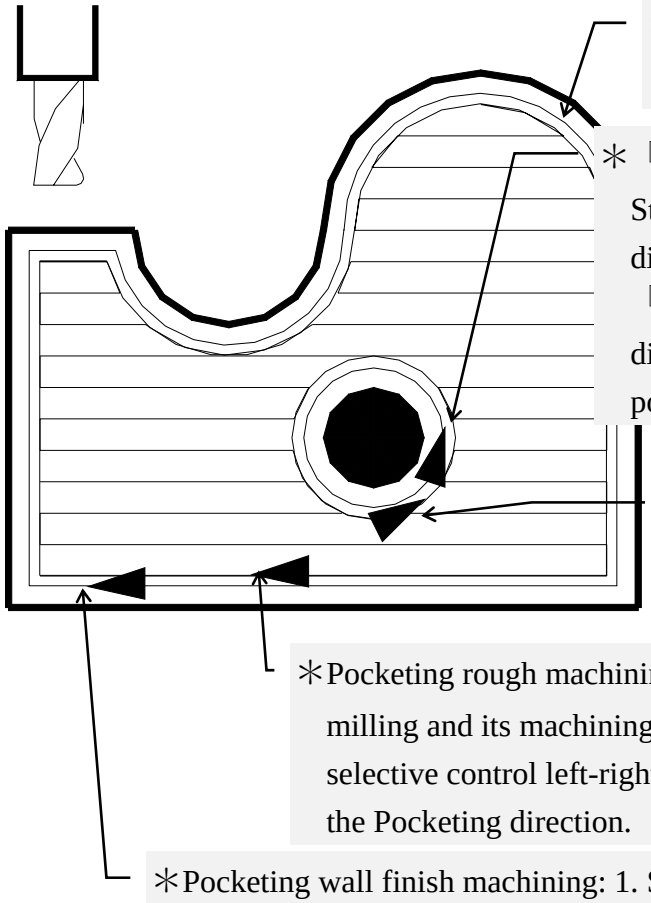
1. **Island Avoidance:** This function can avoid several islands.



2. **Cycle Milling:** 『Machining Depth』cycle milling can be sequentially controlled.



3. Cut Mode ：『rough+finish』,『rough』,『finish』.



＊In between, the space is referred as『Fine machining amount』.

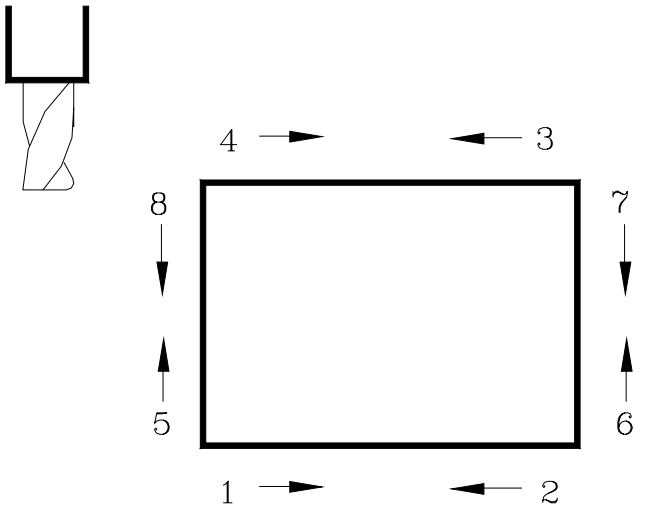
＊『Island Finish Machining』：1. Setting point and Starting point of the tool come from the side direction. The space in between is referred as the 『Finish machining amount』.2. The machining path direction is counter-clockwise. 3. To improve the polish on the surface, user can set Reverse Milling

＊Island rough machining：Operation is selective control shape milling and its machining path is counter-clockwise.

＊Pocketing rough machining：1. First operation is selective control shape milling and its machining path is clockwise. 2. Second operation is selective control left-right milling and its machining path depends on the Pocketing direction.

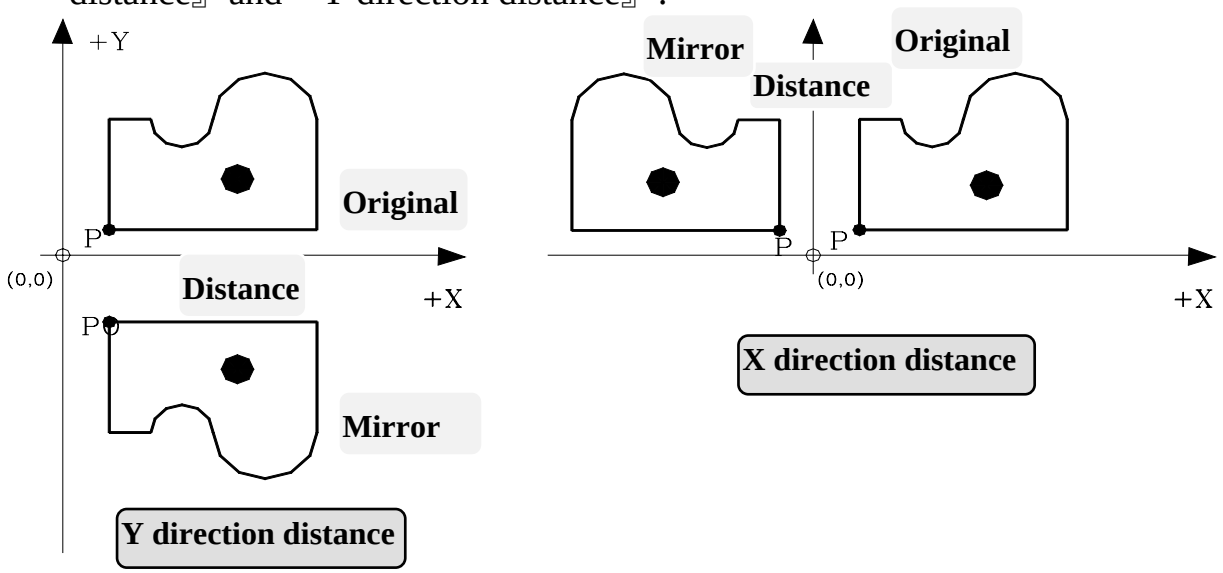
＊Pocketing wall finish machining: 1. Setting point and Starting point of the tool come from the side direction. The space in between is referred as the 『Finish machining amount』. 2. The machining path direction is clockwise. 3. To improve the polish on the surface, user can set Reverse

4. Pocket Direct: When the『Rough』machining is conducted, the direction of the left-right milling can be randomly set from 1 to 8.



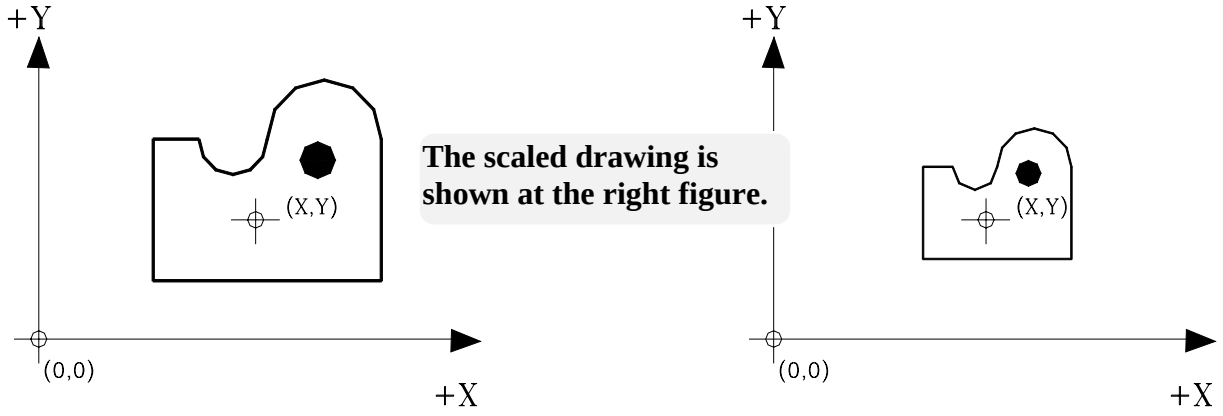
5. **Reverse** (Reverse Milling)**:** For finish machining, you can reverse the machining path to polish the surface of the workpiece.

6. **Image**(Mirror Machining)**:** User can do the mirror projection of the 『X direction distance』 and 『Y direction distance』 .

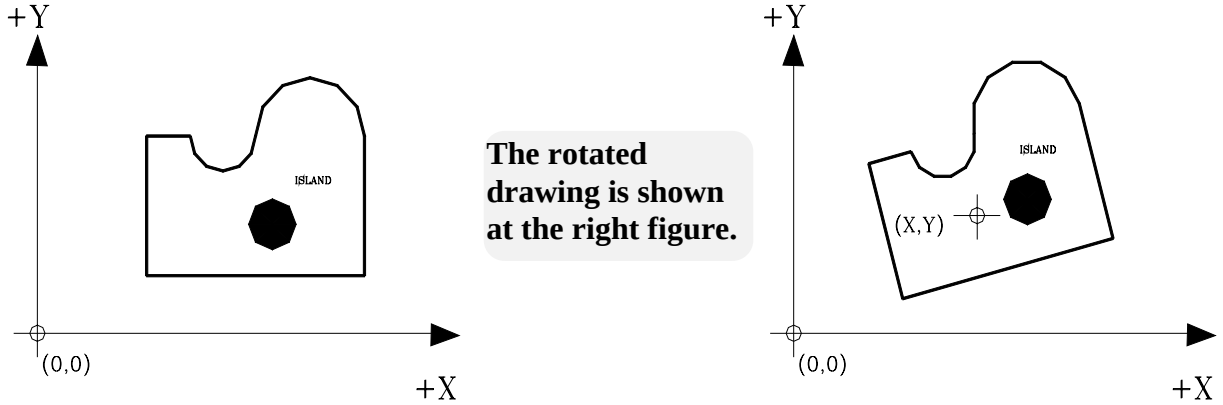


Caution：The mirror projection is based on the 『P point』 of the tool.

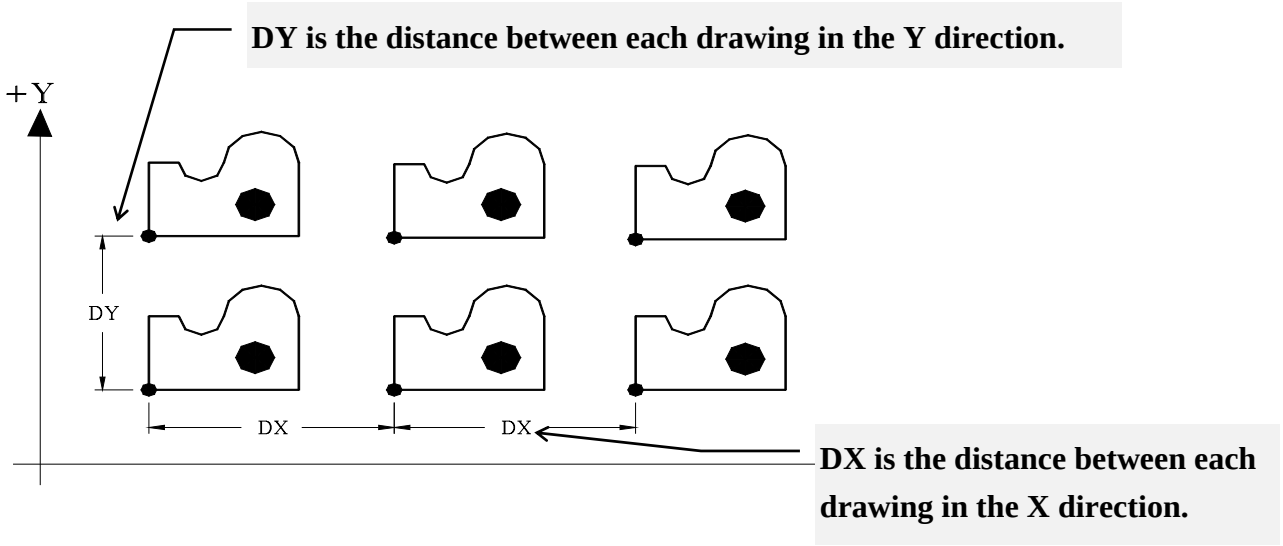
7. **Scale:** User can scale the existing entity. The scale factor ranges from 0.001 to 9.999.



8. **Rotation**(Rotating angle)**:** User can rotate the existing entity.



9. Nx、Ny(Repetitive Machining)**:** User can input one machining drawing to perform multiple machining for the same object by specifying Nx and Ny.


DY is the distance between each drawing in the Y direction.
+Y
DY
DX
DX
DX is the distance between each drawing in the X direction.

Caution：The minimum values of NX and NY were set to 1 and 1, respectively. In the above figure, NX=3 and NY=2.

10. Taper Degree : An angle measured from the vertical, it causes offsetting to be increased as the cutting depth increases in order to produce contours with tapered sides.

11. Taper Finish Option : Finish machining only occurs if the Finish allowance is non zero and the Cut Mode is 1 or 3. If Taper Degree is zero, then this parameter is ignored.

Option	Description
1	The finish machining of the contour takes place at constant Z level, after rough machining at each Z depth, then finish machining.
2	The finish machining of the contour consists of 3 axis XYZ moves. All cutting moves are in the Z direction, at the bottom of each move the tool retracts at rapid to the ***Clearance*** plane and moves in XY to the start of the next move. It then descends to Z zero, and then in three axes to depth. The distance (in the XY plane) between successive cutting passes is controlled by the ***Taper Surface Tolerance*** parameter.
3	Similar to option 2, but the tool cuts in both directions, i.e. It cuts down in three axes to ***Z Final Depth***, moves in XY to the bottom point of the next cutting pass, then move upwards to the Top surface. This produces a much shorted cycle time than option 2, but may not be suitable in all materials.

12. Taper Surface Tolerance : This parameter is used to control the surface finish achieved when the Taper Finish Option is set to either 2 or 3. The distance between successive passes is given by

d = Square root((D * A) – (A * A))
where d = distance, D=ToolDiameter and A is the Surface Tolerance.

13. Pocket Style : It employs 4 different machining strategies when generating a toolpath to machine a pocket. Each has its own virtues.

Style	Description
1	***Uni directional linear machining.*** The outer boundary and any islands are machined as contours, taking into account the ***Finish allowance***. The bulk of the material is then machined away using a series of parallel linear passes. Each pass is distanced from the previous by an amount equal to the ***Tool Diameter*** multiplied by the ***Feed Pitch*** %age. At the end of each pass, the tool is retracted to the Clearance plane and then traversed in XY to the start of the next pass, the tool then descends and begin the next pass. The direction of each cutting pass is specified by the Pocket Angle parameter.
2	***Bi directional linear machining*** This is similar to Style 1, except that the tool is not retracted between passes, instead it follows the outer boundary to the start position of the next pass, and then machines along the pass in the opposite direction to the current pass. This Style produces shorter cycle times than Style 1 but has the disadvantage that the tool is alternately *climb* and *upcut* milling.
3	***Contour machining in towards center*** The Outer boundary, and any Islands are machined by repeated offsetting of the geometry using an offset value which increases by the ***Tool Diameter * Feed Pitch*** %age at each pass. Eventually the offset becomes too big to fit in the pocket, and then it eliminates all redundant moves and outputs the tool path. This style ***cannot*** be guaranteed to perfectly machine all pockets, it is best suited to pockets which are generally convex in shape and do not contain Islands. When used within a suitable pocket shape it produces short cycle times with very efficient tool paths.
4	***Contour machining – out from center*** This Style is identical to Style 3, except that once the tool path has been generated, it is output in reverse order so that the tool starts near the center of the pocket and works outwards. The same comments apply as to the suitability of the style in relationship to the pocket geometry as for Style 3.

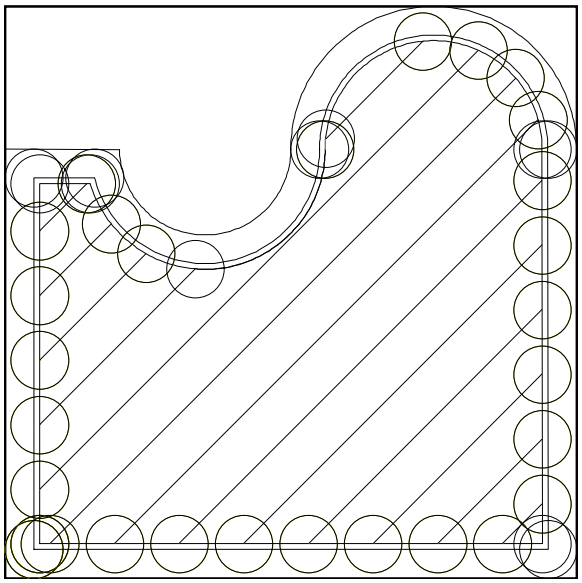
When machining pockets using style 3 or 4, it is possible to utilise a helical drilling motion when descending the tool into the material, eliminating all plunge cutting.

At each point where the tool needs to descend into the material, it drills a hole specified by the *Pocket Spiral R* and *Pocket Spiral Z* parameters.

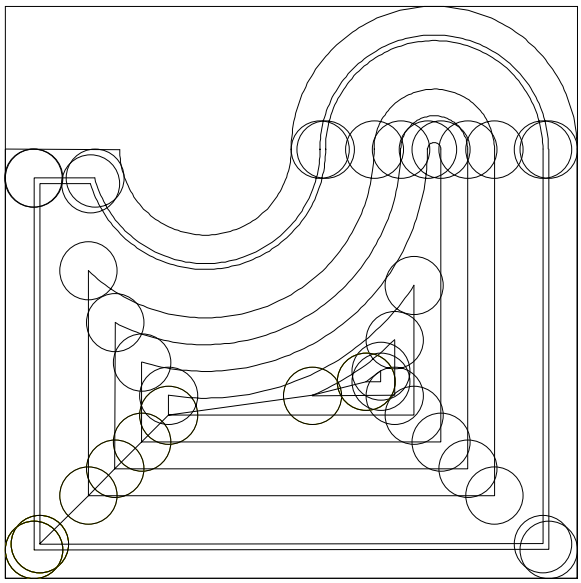
14. Pocket Angle : For pockets machined using Style 1 or 2, this the is angle relative to the X axis of a line, to which all cutting moves are parallel

15. Pocket Spiral R : The radius of the helical drilling motion used when descending the tool into a pocket.

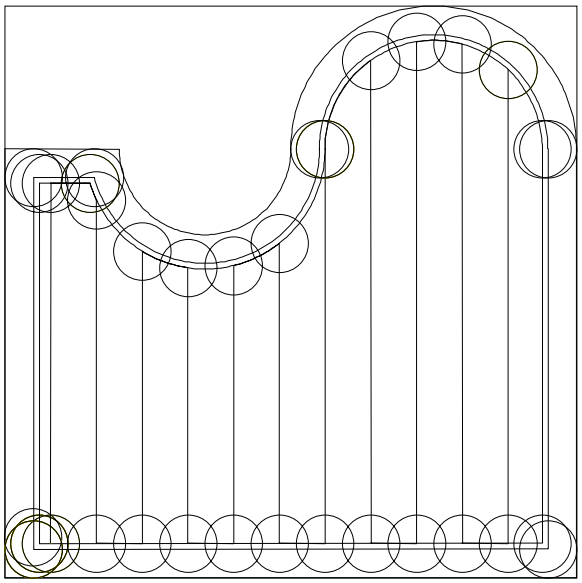
16. Pocket Spiral Z : The pitch of the helical drilling motion used when descending the tool into a pocket.



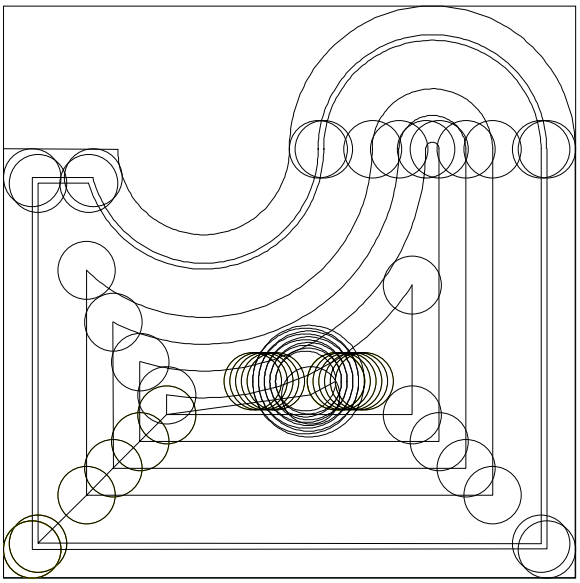
Pocket Style 1, Pocket Angle = 45°



Pocket Style 3.



Pocket Style 2, Pocket Angle = 90°



Pocket Style 4, Pocket Spiral R = 10.0
Pocket Spiral Z = 1.0

17.Changing how machines pockets

When machining Pockets using style 3, it works by offsetting the geometry at increments determined by the **Feed Pitch** parameter, it simultaneously offsets all the contours i.e. the outer boundary and any islands, and then eliminates any overlapping paths. At each offset value the path produced is the same as would be generated for a tool whose radius equals the current offset value. When the offset becomes too large to fit within the boundary, it sorts all the remaining paths to produce the machining program.
This strategy can produce efficient toolpaths for a wide variety of pocket shapes but ***cannot*** be guaranteed not to leave small areas un-machined, as shown in the example below. Usually changing the value of the **Feed Pitch** parameter to a smaller value will eliminate the un-machined areas.

This shape was machined using a 10 mm tool with the **Feed Pitch** set to 80%. The toolpath clearly shows un-machined areas. Changing the **Feed Pitch** to a smaller value corrects the problem. Using a smaller **Feed Pitch** value, creates more offset paths and is less likely to leave un-machined areas.

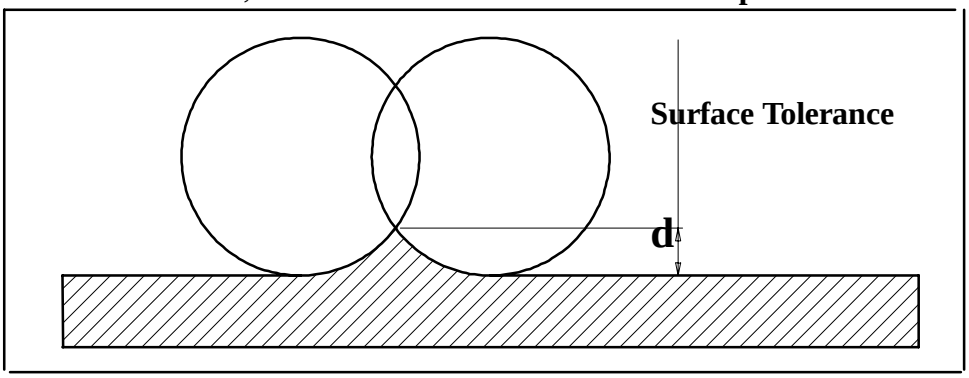
To use this option successfully it is ***imperative*** that the user visually checks the toolpath and confirms that it is acceptable before starting the machine tool. Occasionally the user will decide that this is not the best machining strategy and will opt to machine the pocket using uni-directional or bi-directional linear machining. (Pocket styles 1 and 2).
Contour machining of pockets is most likely to be successful when the (offset) pocket shape does not change significantly as the offset value increases. In the above example the shape changes when the tool can no longer enter the area under the short horizontal line at the top left. When this occurs it is necessary to change the **Feed Pitch** value to a smaller value (about 30%) The shape changes again when the tool cannot get past the island, and effectively becomes two separate shapes, when this happened the **Feed Pitch** value needs to be changed yet again (to about 15%).
In order to assist the user, it can be used to modify the contour machining strategy. It is ignored for all other machining.

18.Controlling surface finish .

When taper machining, the surface finish is controlled by the **Taper Surface Finish** parameter.

```
d = Square root((D * A) – (A * A))
```

where d = distance, D=***ToolDiameter*** and A is the **Taper Surface Tolerance** value.



F1: EDIT Description

1 Input program number 1（The number between 0 to 999）.

＊ Last program number in the brackets.

```
EDIT PROGRAM NO.: 1     (1  )  Coordinate System : 0 (ABS=0,INC=1)
```

Step	Cmd	Fn	X	Y	Z	N/R	N/R	N/R
1	P	0	+50.000	+40.000	+0.000	+0.000	+0.000	+0.000
2	P	0	+50.000	+0.000	+0.000	+0.000		
3	L	0	+100.000	+0.000	+0.000	+0.000		
4	L	0	+100.000	+75.000	+0.000	+0.000		
5	CCW	0	+50.000	+75.000	+0.000	+0.000		
6	CW	1	+20.000	+75.000	+0.000	-15.000		
7	L	0	+0.000	+75.000	+0.000	+0.000		
8	L	0	+0.000	+0.000	+0.000	+0.000	+0.000	-10.000
E 9	L	0	+50.000	+0.000	+0.000	+0.000	+0.000	+0.000

2. “0” means the drawing is based on the Absolute coordinates “1” means the drawing is based on the incremental coordinate.

Fn is a help function code for intersection or tangency.

＊Z value is the starting coordinate value of “ machining depth”, it is always represented by absolute coordinate.

```
F1:INS   F2:DEL   F3:PG/UP   F4:PG/DN   F5:DATA/END   F6:CHK/DRAW   {>:QUIT
```

Note : Function key define

F1:INS **:** Insert a column **F2:DEL** **:** Delete a column

F3:PG/UP **:** Page up **F4:PG/DN** **:** Page down

F5:DATA/END **:** Set end column

F6:CHK/DRAW **:** Cursor moves to any position, then press F6 to the pocketing graph inspection screen. Check whether this is correct and machining is executed at this program line.

＊Press the Quit key to go back to the input screen.

Caution：1 For pocketing, the machining draw must be a close loop. So does the island.

2.The first setting point P is for the pocketing. The rest of the setting points are for the island machines and every island machining must have a setting point P.

Parameter setting:

```
 EDIT PROGRAM NO.: 1     (10 )  Coordinate System : 0 (ABS=0,INC=1)

Tool Diameter          :    10.000      Nx :     1          Ny :      1
Feed Pitch (10%-100%)  :    75          Dx :    +0.000      Dy :     +0.000
Z Final Depth          :    10.000      X&Y Feedrate           :   5555
Z Cutts                :    10.000        Z Feedrate           :   5555
Cut Mode(1=R+F,2=R,3=F):     1         Sleeping (0=Off,1=On)  :      030
Finish                 :     0.000     Display (0=XYZ,1=XY)   :      0
   Reverse (0=Off,1=On):     0         Tapper Degree (0-85)   :     +0.000
Safe Height            :     5.000         Finish option(1-3) :      1
Clearance              :     2.000         Surface tolerance  :      0.050
Image(0=N,1=X,2=Y,3=XY):     0         Index No.              :      1
      Distance X       :    +0.000     Tool No. (T)           :      0
      Distance Y       :    +0.000     Tool Dia.   Compen No. :      000
Scale         (X,Y)    :     1.000     Tool Length Compen No. :      0
              ABS.X    :    +0.000     Spindle RPM            :   +500
                  Y    :    +0.000     C Axis DTH             :     +0.000
Rotation      (X,Y)    :    +0.000     C Axis STH             :     +0.000
              Dth      :    +0.000     Pocket Style  (1-4)    :      4
              Times    :     1                Angle           :     +0.000
              Mode     :     0                Spiral R        :      2.000
  ABS.X0:    +0.000  Y0:    +0.000             Spiral Z        :      2.000
  ABS.X1:    +0.000  Y1:    +0.000
```

＊Press the QUIT key to go back POCKET screen.

Parameter description:

Tool diameter : The diameter of the tool to be used, must be greater than zero.

Feed Pitch : The overlap of the tool between 1 cutting pass and the next 10-100%.

Z final depth : The final depth of the contour

Z Cutts : The incremental z depth. If Z Cutts is less than Z final depth, it produces a number of passes in Z, at increments of Z Cutts in order to achieve the final depth. The final pass may be less than Z Cutts.

Cut Mode : 1 = Roughing + Finishing, 2 = Roughing only, 3 = Finishing only

Finish : Finish allowance for final machining. The Finish machining will only take place if this parameter is non zero, and Cut Mode specifies either 1 or 3

Reverse : 1 = Reverse direction of finish cut, 0 = finish cut is in the same direction as the geometry.

Safe Height : The assumed Z value of the tool at the start of the program, Z position for retract at end of program

Clearance : The Z value at which Feedrate changes from Rapid to Feed when the tool is descending.

Image : Mirror image flags, 0 = None, 1= Mirror about X, 2= Mirror about Y, 3= Mirror about X and Y.

Distance X
Distance Y

Incremental distance from Geometry origin to mirror axis

Rotation (X,Y) : Rotation angle in the XY plane.

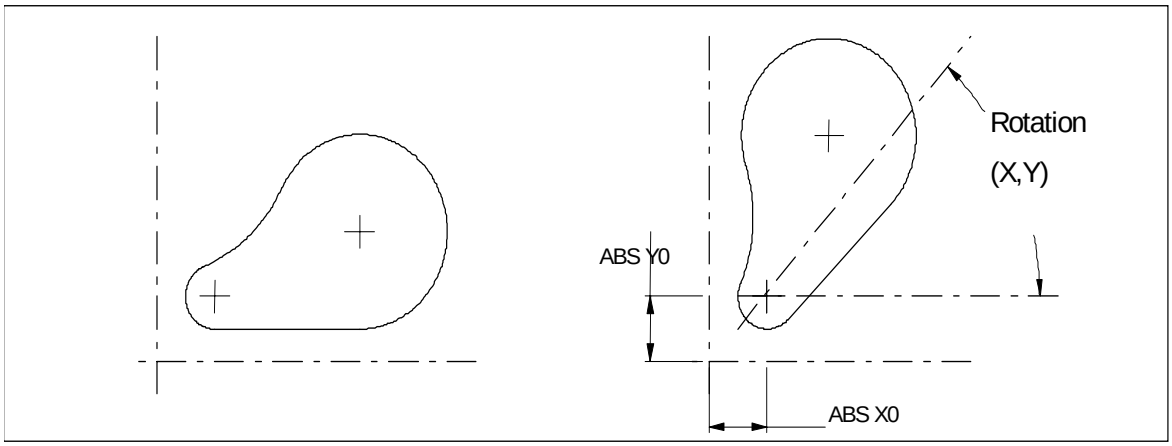
Rotation Dth : Rotation incremental angle positive counter clockwise.

Rotation Times : Number of Copies

Rotation Mode : If 1 the geometry is rotated by the angle specified above. If 0 then the geometry is translated to a new position according to the angle and radius from the rotation center.

ABS.X0 ABS.Y0

These two parameters define the XY centre of rotation used by the Rotation(X, Y) angle.

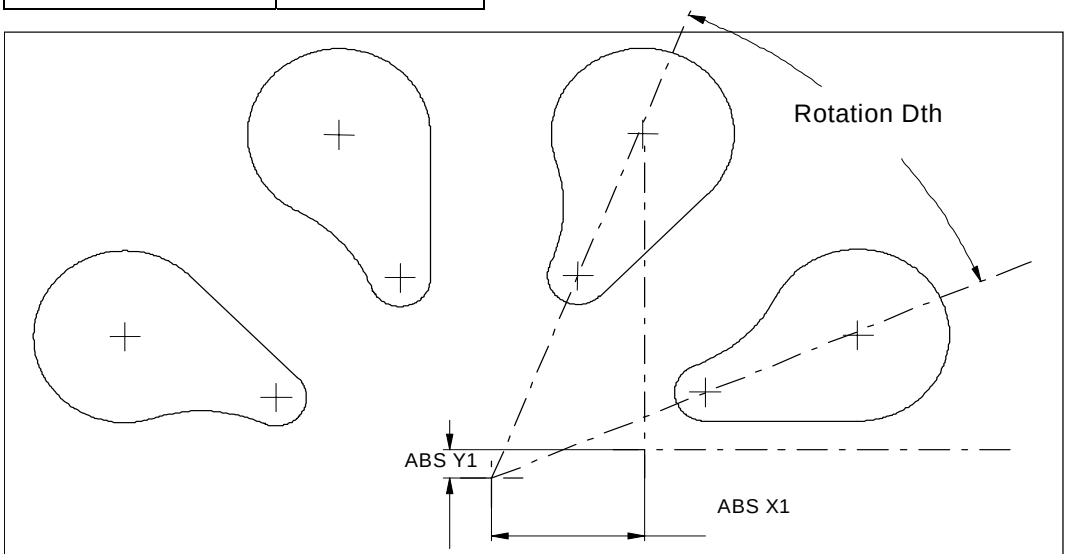


ABS.X1 ABS.Y1

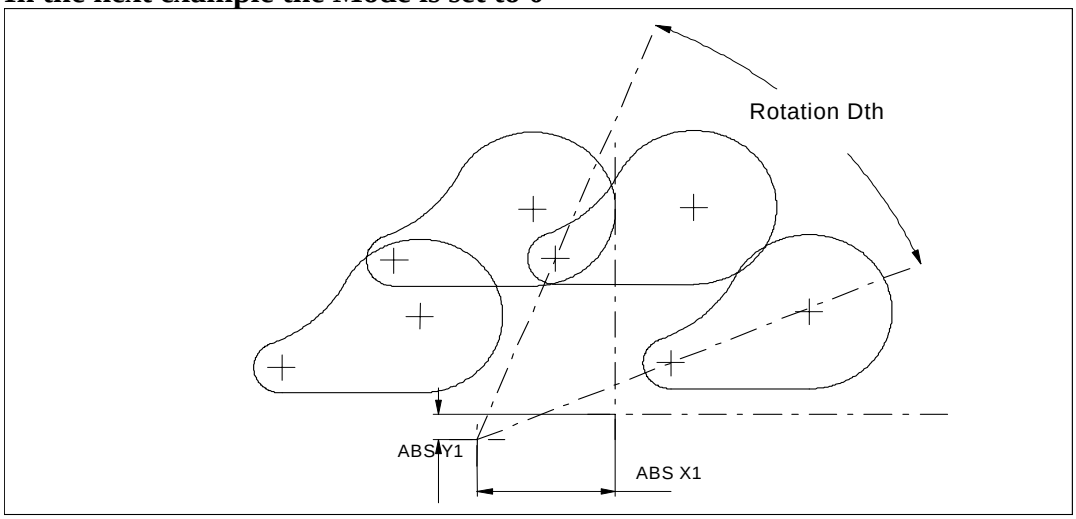
These two parameters define the point about which the geometry is rotated when producing multiple copies around an arc.

In this example the parameters are …

Rotation Dth	**45**
Times	**4**
Mode	**1**
ABS X1, Y1	**-40, -10**



In the next example the Mode is set to 0



XY Feedrate : FeedRate for all XY (and XYZ) moves

Z Feedrate : Feedrate used when plunging into the workpiece.

Tool No. (T) : Tool number used in tool change block.

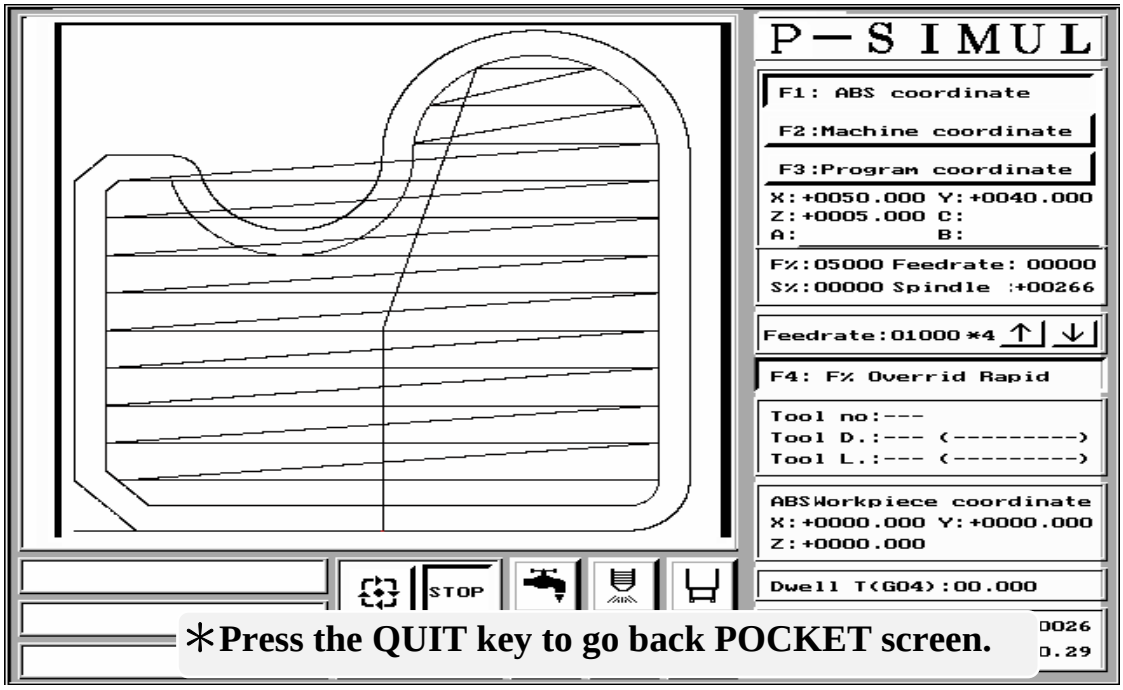
Tool Dia. Compensation No.
Tool Length Compensation No.
Tool length and diameter compensation numbers.

Spindle RPM : Spindle speed

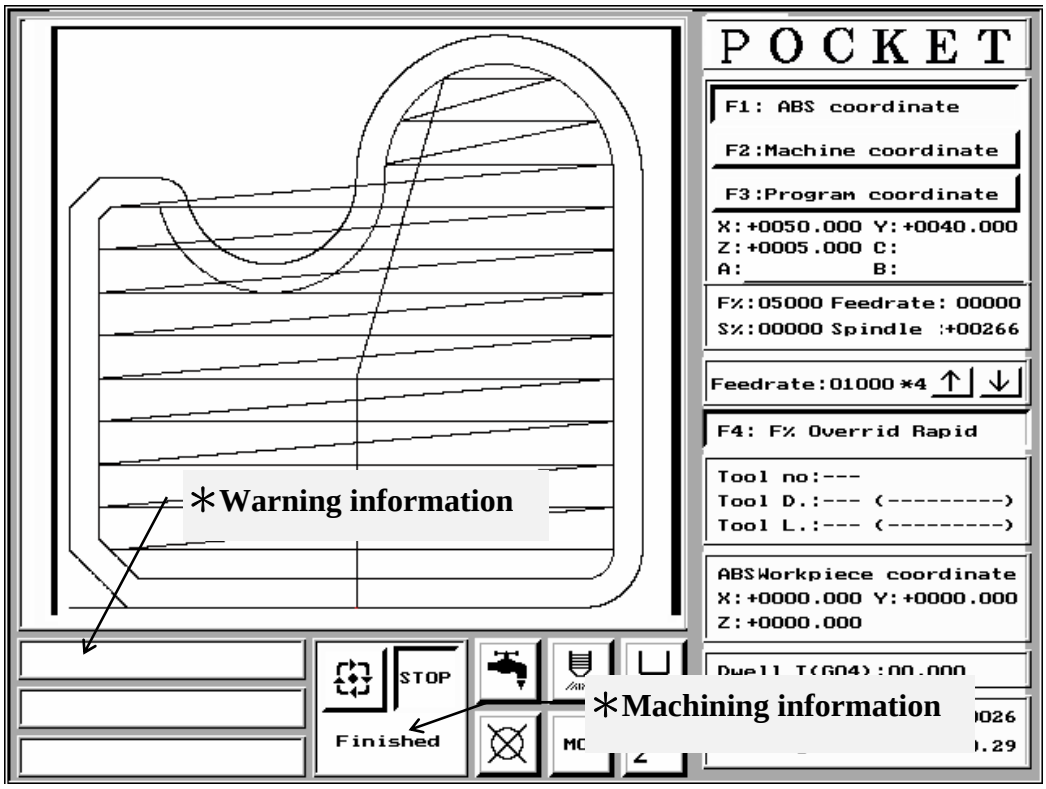
C Axis Dth. : Incremental C axis angle. If non zero, the NY parameter is assumed to refer to the number of repeats in C axis.

C Axis Sth. : Initial C axis angle.

F2: DRAW
Description:

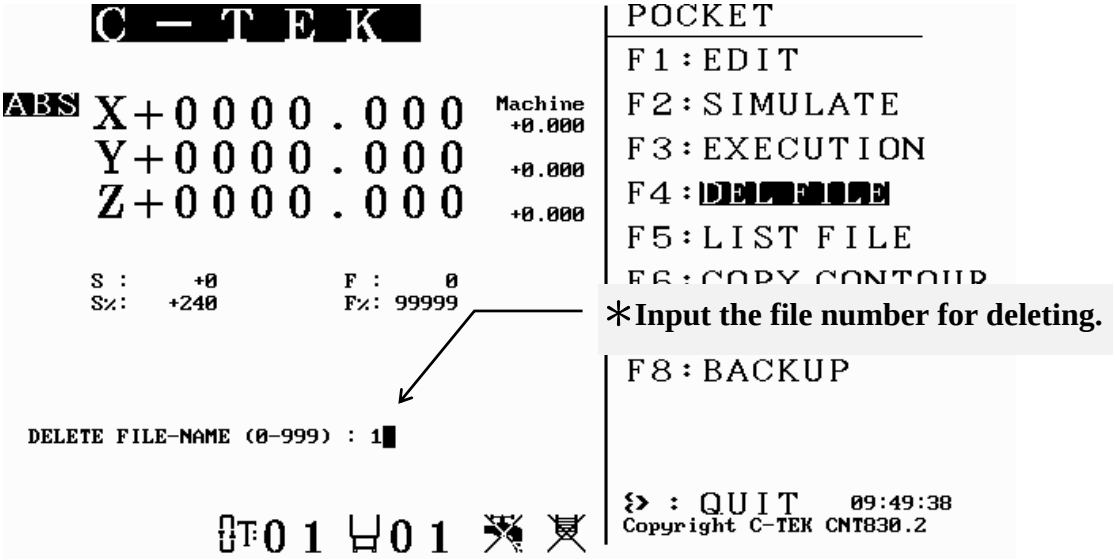


F3: EXECUTION
Description:



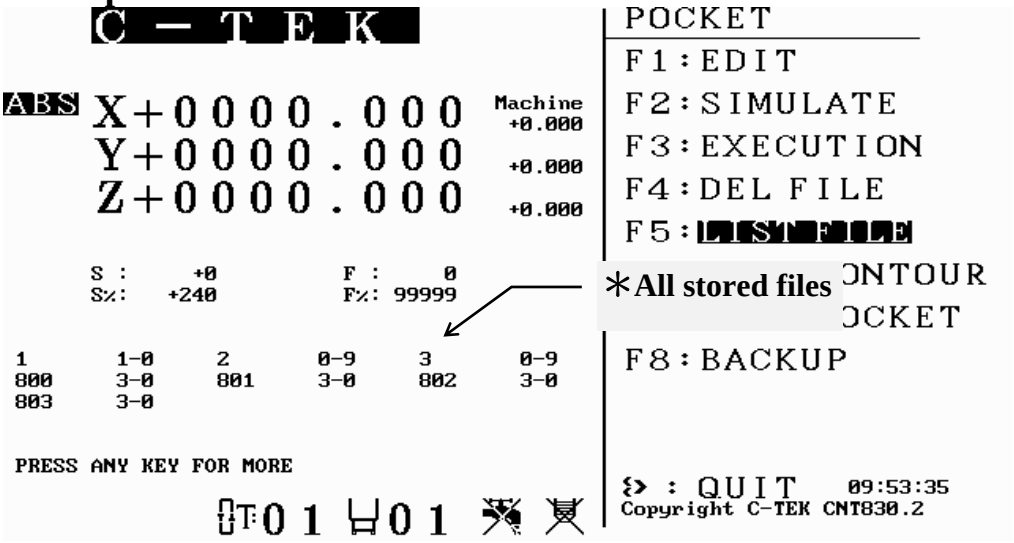
＊**Press the stop button on the control box to stop the machining. The manual function on the control box can be used to check the workpiece. When the continuation operation is pressed, the machining operation is continued.**

F4: DEL FILE
Description:



F5: LIST FILE

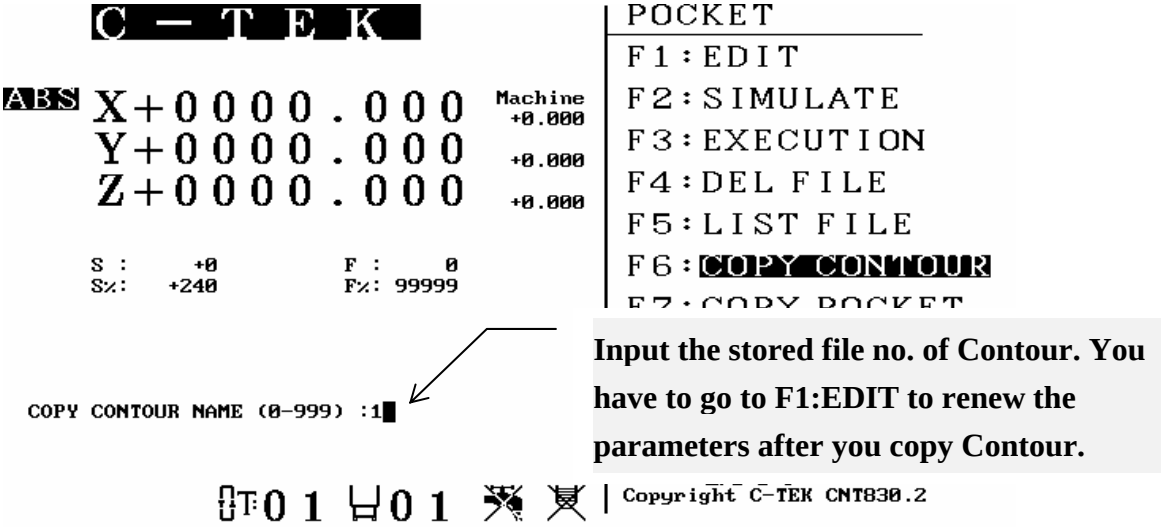
Description:



＊All stored files

F6:COPY CONTOUR

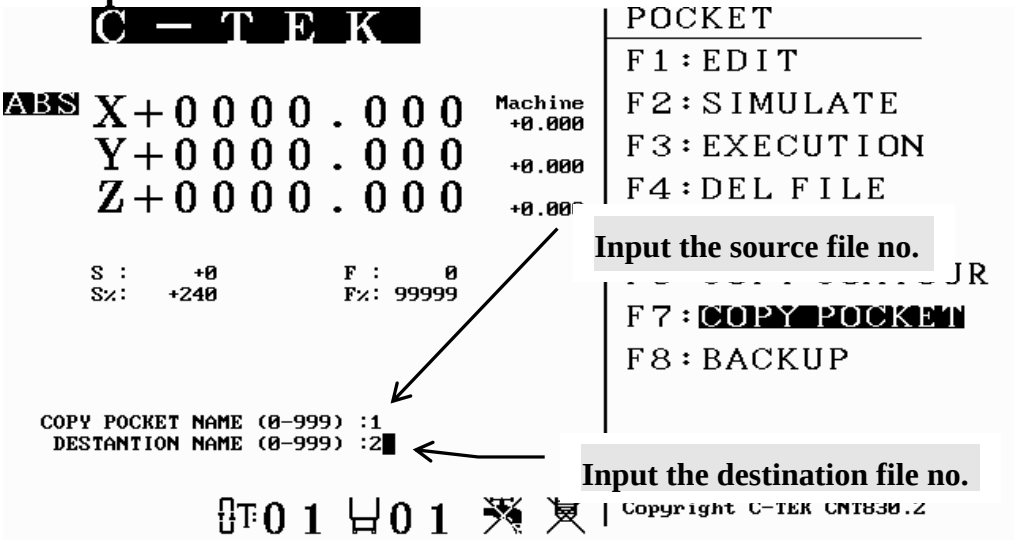
Description:



Input the stored file no. of Contour. You have to go to F1:EDIT to renew the parameters after you copy Contour.

F7: COPY POCKET

Description:

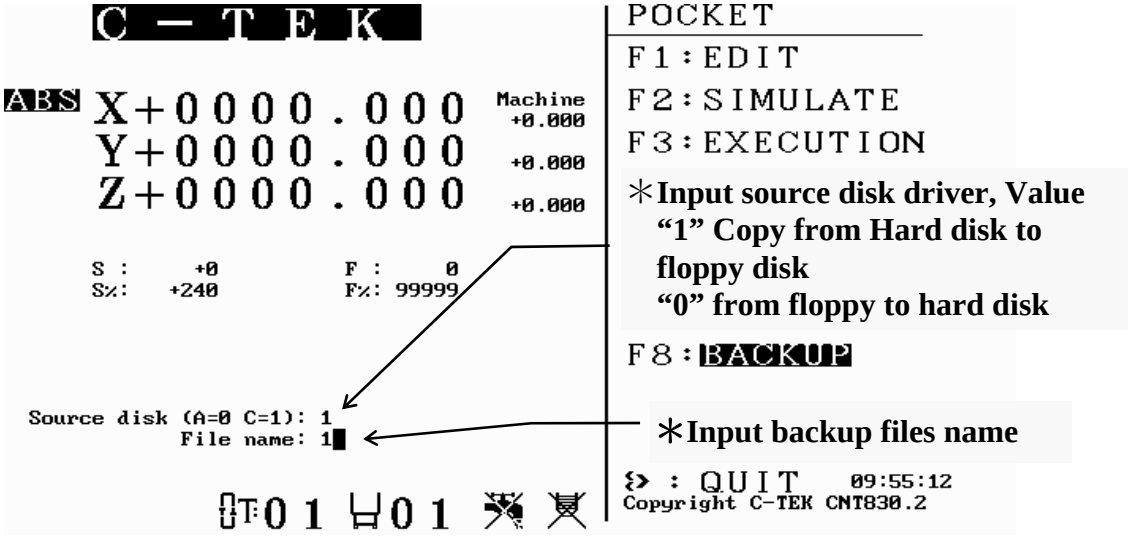


Input the source file no.

Input the destination file no.

F8: BACKUP

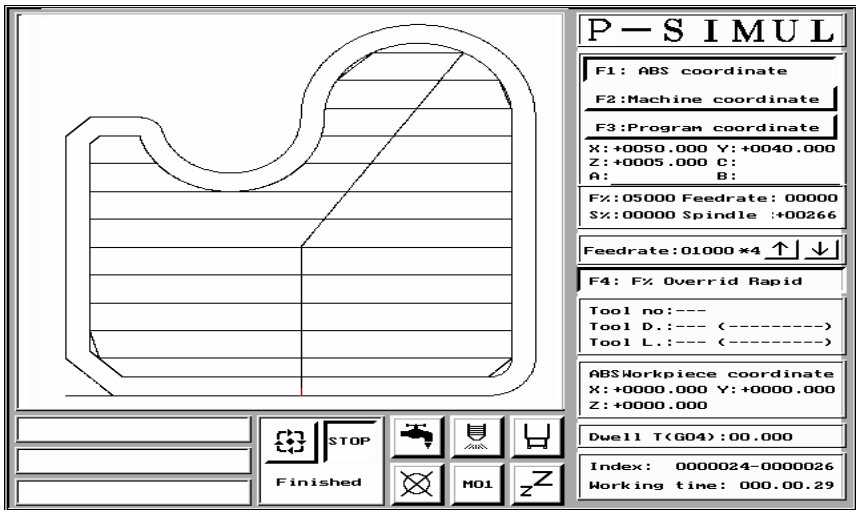
Description:



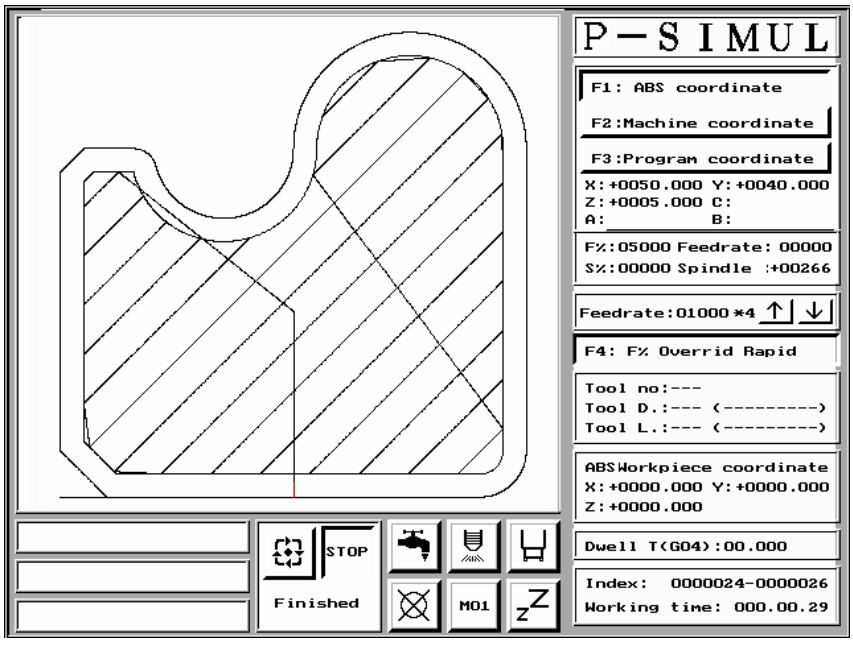
＊**Input source disk driver, Value**
“1” Copy from Hard disk to floppy disk
“0” from floppy to hard disk

＊**Input backup files name**

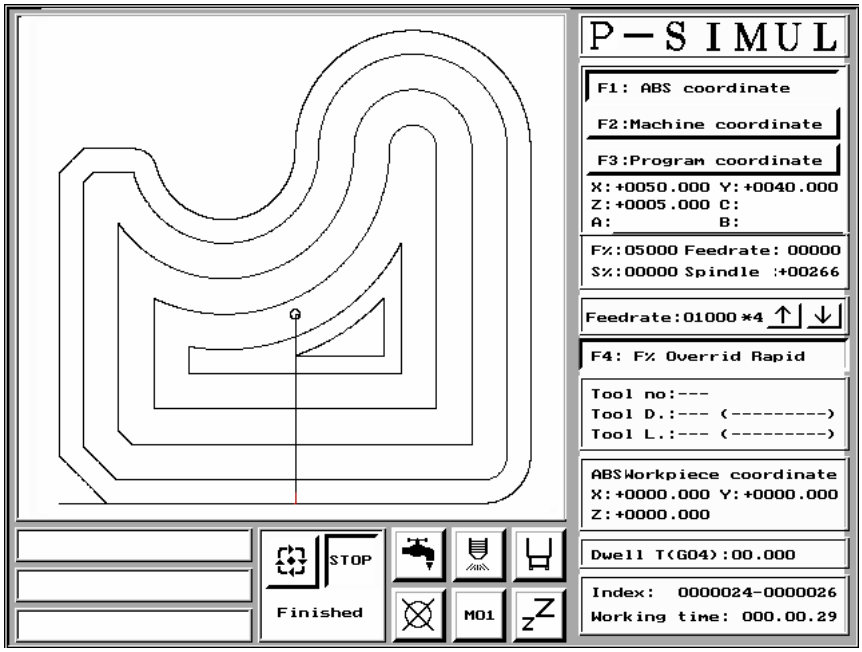
Sample edit program with pocket style=2, angle=0



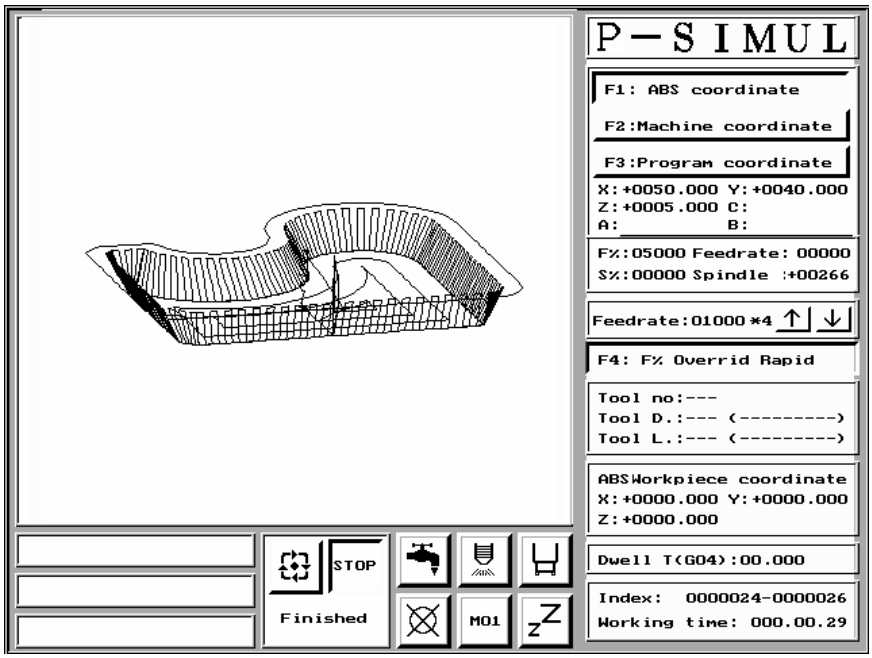
Sample edit program with pocket style=2, angle=45



Sample edit program with pocket style=3



Sample edit program with pocket style=3, taper=30, Taper Finish Option=3



12.Geometry help function

Cmd

Short for Command, Cmd defines what the step represents. It is one of **P**, **L**, **CW** or **CCW** meaning a **P**oint, a **L**ine, a **C**lock**W**ise arc or a **C**ounter **C**lock**W**ise arc.

Fn

Short for Function number. The function number associated with each Command instructs how the remaining data should be interpreted in order to define Point, Line or Arc. The default function number is Fn0.

X, Y, Z

The end point in cartesian coordinates of the Point Line or Arc being defined.

I, J

The centre point of an arc step measured incremental from its start point. Ignored for Points and Lines.

R+/ C-

Instead of specifying an arc's centre point using I and J values, its Radius can be specified instead. If I, J and Radius all contain values, then I and J are ignored.

If The current step is a **Line** the Radius / Chamfer column specifies the size of a Fillet(R+) or Chamfer(C-) to be constructed between this step and the next.

The geometry editor screen layout.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1								

The geometry editor presents as a grid displaying one column for each parameter. Not all columns will contain useful data in each step. For example The I and J columns do not normally contain useful data if the current step is a Line or a Point Command.

The combination of **Cmd** and **Fn** instructs the system how to interpret the rest of the data in each step. Using Fn0 – the default, the data is in fact interpreted as described above, and as suggested by the column headings displayed in the geometry editor. The X, Y and Z values represent the end point of each step (and therefore the start point of the next step), The I and J columns are used to define arc centre points – on CW and CCW steps. The last column is used to defined either a Fillet or a chamfer which is to be constructed between the current step (if it is a **L**ine segment only) and the next. If the current step is either CW or CCW, then the Fillet/Chamfer column defines the radius of the arc step.

It is important to remember that apart from **P**oint steps which can be defined in a single step, **L**ine and **CW / CCW** steps define only the end point of the line or arc. The start point of the line or arc is defined by the previous step. In other words each step defines the end of one line or arc and the start of the next. There can be no gaps in the shape being defined.

General Specification

The following features are supported: -

- The user may freely switch between programming arcs using either Radius values (where both the start and end point of the arc are known), or I & J values to explicitly define the arc center.
- Automatic insertion of ***Fillets*** or ***Chamfers*** between one block (Step) and the next.
- Calculation of ***intersection*** points between one block and the next. E.g. A linear step can be defined as being from the current point at a specified angle and ending at the intersection point with the following line or arc step.
- Calculation of ***tangent*** lines and arcs. E.g. a Line step can be defined as starting from the current point, and ending at its point of tangency with the following arc.
- Polar programming. It can accept polar coordinates for X, Y, I and J coordinates (although I and J are likely to be zero if the Polar origin is the centre of the circle). Polar programming is switched on and off using Function codes attached to a **Point** step.

The Point functions

It can freely switch between Cartesian (X Y coordinate programming) and Polar (Radius and Angle) coordinate systems using functions attached to a **Point** step

Function	Description
0	• Defines the start point of a contour in the ***current coordinate system*** • ***Function 0 defines a move point. I.e the start of a contour.***
1	• Defines an *auxiliary point* in the current coordinate system. An auxiliary point is one which acts as a temporary start point for the following Line or Arc step. Auxiliary points are not (normally) on the contour.
2	• Defines the ***absolute*** position of a new Polar coordinate system origin in the ***current coordinate system***. • The Polar coordinate system is switched ON. • Until the Polar coordinate system is switched off again, the values in the **X** column represent ***Polar Radius***, and the values in the **Y** column represent ***Polar Angle***, polar coordinates are always absolute. • I and J values are also affected so that I and J are the absolute polar coordinates of a circle centre measured from the Pole. • The headings for X and Y columns should change to remind the user that polar programming is in effect. (e.g. use PR for radius and PA for angle)
3	• Defines the ***incremental*** position of a new Polar coordinate system origin in the ***current coordinate system***. • Programming **P, Fn3, X0, Y0** makes the current point i.e.the end of the previous step, the new polar origin.
4	• The Polar coordinate system is switched **OFF**. • Programming continue in Cartesian coordinates (X, Y) using X0 Y0 as the absolute datum.
• **Function 1,2,3,4 does not define a move point**.	

Line functions.

Function	Description
0 or 1	Line starts at the end-point specified in the previous step, and ends at the point defined in the X and Y columns. Can be used to specify that the Radius column contains a Fillet/Chamfer value, it is supported for compatibility with arc functions but it is not strictly necessary.
2 or 3	Specifies a line **starting** at a fixed point i.e. the end point of the previous step, and **ending** at an intersection point with the following line or arc step. The values in the X and Y columns define a point on the line, which may or may not be its endpoint. If the following block is a Line step then functions 2 and 3 are interchangeable and the following line **must** use function 4 or 5. But if the following block is an arc, then function 2 will select the *Left Hand* intersection and function 3 will select the *Right hand* intersection.
4 or 5	Specifies a line which passes through a fixed point, defined in the X, Y columns, at a fixed **angle** to the X axis. (The angle is specified in the I column.) The point specified in the XY columns may be either the line start point, **or** its end point depending on the context in which the line appears. If the line is the first of a pair of intersecting elements, then the X&Y values are the start point of the line, and its end point is where it intersects the following line or arc. If the line is the second of a pair of intersecting elements, then the X and Y values are the end point of the line, and the line starts at the point where it intersects the previous line or arc.
6	Specifies a line **starting** at a fixed point i.e. the end point of the previous step, and **ending** at its point of tangency with the following CW or CCW step. The tangent point is chosen to preserve continuity of direction from the line to the arc.
7	Specifies a line **ending** at a fixed point entered in the X&Y columns, and starting at its point of tangency with the previous CW or CCW step. The tangent point is chosen to preserve continuity of direction from the arc to the line.
8	Specifies a line tangent to both the preceding and the following arc steps. The points of tangency are chosen to preserve continuity of direction from the first arc, via the line to the second arc. XYZIJR are all ignored
9 or 10	Fn9 defines a line at a fixed angle tangent to the arc in the next step. Fn10 defines a line at a fixed angle, tangent to the arc in the previous step. Both Fn9 and Fn10 calculate the tangent point with the arc, and then change the line function to FN4.

- If an angled line, or a tangent line is followed by an arc, then the arc centre point **must** be programmed using I and J. Since the arc start point is unknown, it is assumed temporarily to be a 360 degree arc, therefore the I and J values may be programmed incremental from the end point instead of the start point.
- Similarly if an arc is followed by an angled line, or a tangent line, then it too must have been programmed using I and J to define its centre point and radius, since the arc's true end point is unknown, it should be programmed as a 360 degree arc.
- Lines defined using Functions 1 to 7 may include a Fillet / Chamfer value.
- To determine the left hand and right hand intersection between a line and arc, rotate the line by 90 degrees counter clockwise about a point mid way between the two intersections, now sight along the line to determine right and left.

Arc functions

Function	Description
0	The arc starts at the point defined in the previous step, and ends at the point defined in the X and Y columns. If the Radius column contains a non-zero value it is assumed to be the arc radius, otherwise the arc centre is defined (incremental from the start of the arc) using I and J parameters..
1	Specifies that arcs are defined by I and J, and that the Radius column contains a Fillet / Chamfer value. I & J are from the start of the arc.
2 or 3	Specifies an arc that **starts** at a fixed point i.e. the end point of the previous step, and **ends** at its intersection with the following line or arc. By implication the following line or arc starts at the same intersection point, and ends at its programmed XY position. The arc must be programmed using I and J (incremental from the **start point**) to define the centre point, and is initially assumed to define 360 degree arcs. If the following element is a line, it must be programmed using function 4 or 5 i.e. a Line at a specified angle, and the XY value is its end point. If the following element is an arc, then its centre must be programmed using I and J.
4 & 5	Specifies an arc which ***starts*** at a known point and is tangent to the Line or Arc step which follows. The XY columns contain the start point of the arc and the Radius column contains its radius. The I and J values are ignored. If two solutions are possible, Function 4 selects the minor arc and Function 5 selects the major arc. If the following step is an arc, then a negative value in the Radius column will cause one arc to be ***inside*** the other, while a positive value will ensure that the two arcs are both ***outside*** each other.
6 & 7	Specifies an arc which is tangent to the Line or Arc step which precedes it, and ***ends*** at a known point and. The XY columns contain the end point of the arc and the Radius column contains its radius. The I and J values are ignored. If two solutions are possible, Function 4 selects the minor arc and Function 5 selects the major arc. If the preceding step is an arc, then a negative value in the Radius column will cause one arc to be ***inside*** the other, while a positive value will ensure that the two arcs are both ***outside*** each other.
8 & 9	Specifies an arc tangent to both the previous and the next elements. Only the Radius column has data.
10	Defines an arc whose I&J values are incremental form the **end** point i.e. the center is relative to the values in the X and Y columns and not the end of the previous block. It is typically used to define an arc whose start point is unknown because the start of the arc is found from an intersection or a tangent with the previous block.
11	Defines an arc, with **unknown centre point** and radius. The arc starts at the end of the previous Line or arc, and which ends at the point defined in the X and Y columns. The arc is tangent to the previous Line or arc.
12	Defines an arc, with unknown centre point and radius. The arc starts at the end of the previous Line or arc, and which ends at the point defined in the X and Y columns. The arc is tangent to the following Line or arc. The values in the I, J and Radius columns are all ignored.

- Arc function 2 and 3 will normally use I & J measured from the start point, but if this is not known (because the previous step specified an end point defined by intersection), then I & J are assumed to be from the end point.
- To determine the left hand and right hand intersection between a line and arc, rotate the line by 90 degrees clockwise about a point mid way between the two intersections, now sight along the line to determine right and left.
- To determine left hand and right hand intersections between two arcs, sight along a line which goes from the centre of the first arc to the centre of the second arc.

Use of I & J with Arc functions

The following table describes how the use of I and J values depend on the arc function number and the previous step.

Arc Function	I and J settings	R values
0	Always incremental from the **start** point of the arc. i.e. the XY coordinate in the previous step.	Specifies radius of arc, I and J are ignored
1	Always incremental from the **start** point of the arc. i.e. the XY coordinate in the previous step.	Specifies a fillet or chamfer.
2 or 3	If the previous step is a **move**, or a **line** using function 0, 1 or 7, or it is an **arc** using function 0, 1 or 10, then the value in the XY column of the previous step defines its end point and therefore the start point of the current arc. I and J are incremental from the **start** point of the arc. i.e. the XY coordinate in the previous step. For all other line and arc functions in the previous step, the end point is not known and therefore the I and J values in the current step are taken to be incremental from the arc **end** point. I.e. the XY coordinate in the current step.	Specifies a fillet or chamfer.
4, 5, 6, 7, 8 and 9	Ignored	Defines the arc radius (can be negative)
10	Always incremental from the **end** point of the arc. i.e the XY coordinate specified in the arc step.	Specifies a fillet or chamfer
11	Ignored	Specifies a fillet or chamfer
12	Ignored	Ignored

Geometry Groups

As far as system is concerned, geometry *i.e. the description of a Contour or a Pocket's shape as a series of connected line and arc segments*, can be described under two group headings …

Simple Geometry

Where the start and end of each line or arc in the shape can be easily deduced from the component drawing.

And …

Complex Geometry

Where the end points of the lines and arcs defining the shape cannot easily be deduced from the drawing, and which need to be calculated using various geometric functions.

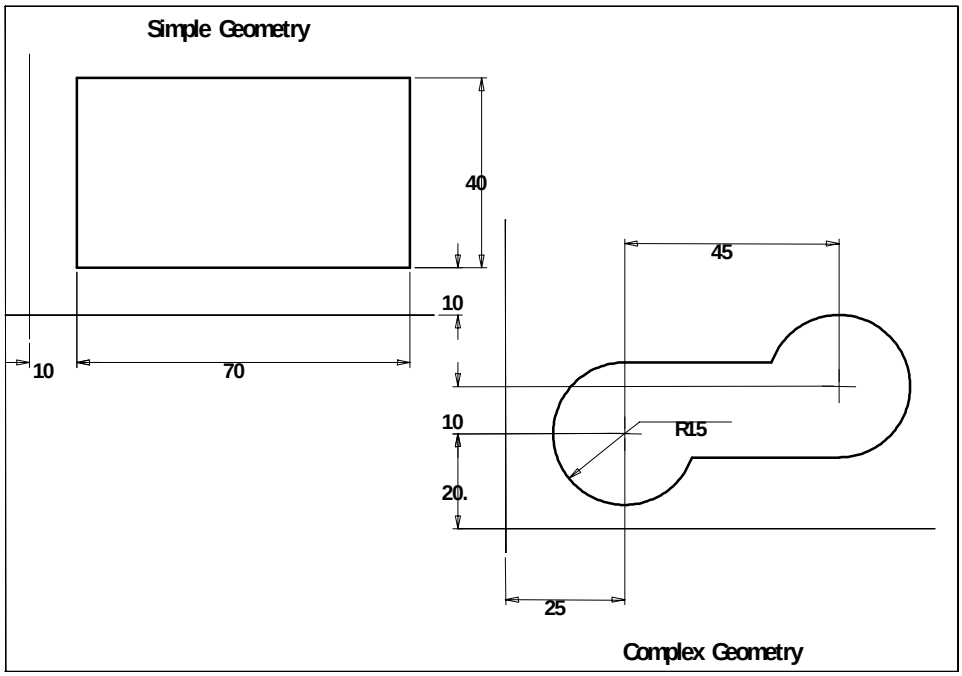


Figure 0. Examples of Simple and Complex geometry

The component on the left is easily defined. The end points of each line can be deduced from the drawing without need for calculation.

The component on the right is more complex. Only one end of each of the lines and arcs can be read from the drawing. The other end must be calculated as the intersection between a line and an arc.

Simple geometry

As stated earlier ***simple geometry*** does not require any calculations to find the end point of each line or arc comprising a contour or pocket definition. In system terms this means that the shape can be programmed using the basic **P**oint, **L**ine **CW** and **CCW** arc steps together with Function 0.

*When using **Fn0**, arcs can be defined either using I and J to define the centre point, **OR** using Radius column to define the arcs radius (in which case the system calculates the arc's centre point). Any line segment – L step - can include a value in the Radius column which defines either a Fillet[1] or a Chamfer[2] to be constructed between the current geometry segment and the next. Fillets cannot be attached to arc steps. A **P** step defines the start point of a 2d profile in cartesian coordinates.*

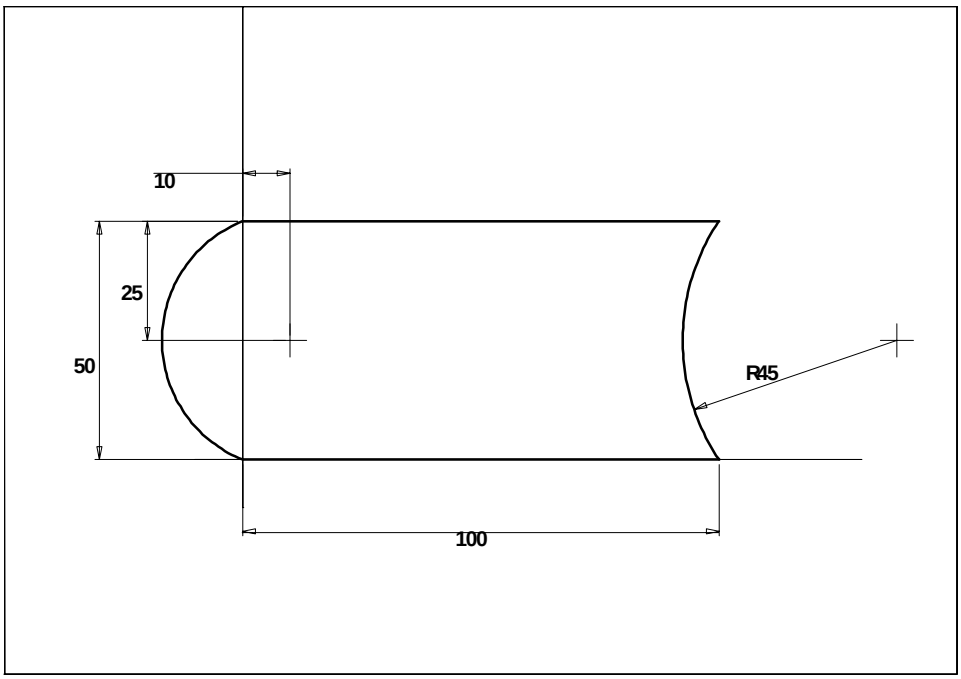


Figure 1, a simple profile.

Programming this shape is straight forward because the end point of each line and arc segment can be easily deduced from the component drawing.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	100.000	0.000	0.000	0.000	0.000	0.000
3	CW	0	100.000	50.000	0.000	0.000	0.000	45.000
4	L	0	0.000	50.000	0.000	0.000	0.000	0.000
E 5	CCW	0	0.000	0.000	0.000	10.000	-25.000	0.000

Program 1

- The shape is programmed in a generally counter clockwise direction.

[1] A fillet is an arc of fixed radius, tangent to the current geometry segment and the next. The system will calculate the fillet's start, centre and end points and its direction.

[2] A Chamfer is a short line segment constructed between the current geometry segment and the next. The system calculates both the start and end points of the chamfer.

Fillets and chamfers

Consider this new component …

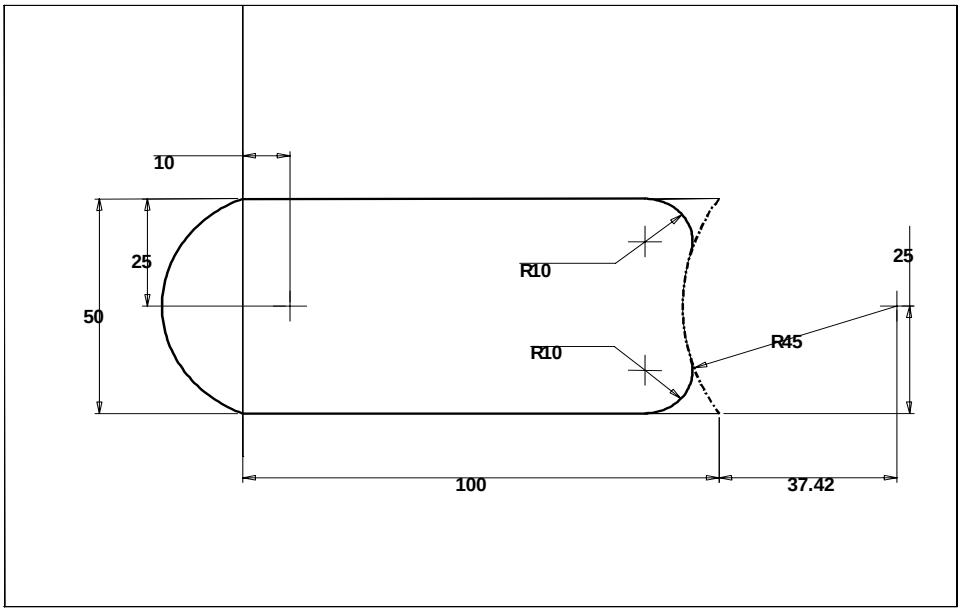


Figure 2.

This new shape might appear to be an example of *Complex geometry*, but this is not really the case. If the R10 fillets are ignored, then this is eactly the same shape as Figure 1. There is even some new information, the drawing now defines the centre point of the R45 arc.

As stated earlier, if the current step is a ***line***, then a fillet or chamfer can be attached to that step simply by specifying its size in the radius column. What we want to do is to program a fillet on a CW or CCW step. As things stand it can not be done, as soon as a value is entered into the Radius column, the system assumes that it is the radius of the CW / CCW step that is being programmed. It is necessary to tell the system that the radius value defines a fillet or a chamfer, ***this is done using arc function 1.***

Fn1 specifies that arcs shall be programmed using I and J values to define the arc centre, ***and*** *that the Radius column defines either a Fillet if the Radius value is positive, or a chamfer if the radius value is negative*

The program now looks like this …

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	100.000	0.000	0.000	0.000	0.000	10.000
3	CW	1	100.000	50.000	0.000	37.420	25.000	10.000
4	L	0	0.000	50.000	0.000	0.000	0.000	0.000
E 5	CCW	0	0.000	0.000	0.000	10.000	-25.000	0.000

Program 2

There are several important things to note about this new program.

- The line at step 2 is still defined as ending at X100 Y0, even though the actual end point is not known. The Radius column has a value of 10, indicating a 10mm radius fillet is to be constructed between this line and whatever appears in step 3
- The arc at step 3 now uses **Fn1** to specify that its centre is defined by I and J values. The I and J values are taken from the start of the arc – ignoring the fact that there is a R10 fillet in between steps 2 and 3. The radius column can now be used to specify the R10 fillet in the top-right corner of the shape.

- The end point of the arc in step 3 has not changed. It will be re-calculated by the system when the fillets are constructed.
- The arc in step 5 is still defined using Fn0. The value in the Radius column is zero, and so the system uses the I and J values. If the Fn0 were to be replaced by Fn1, the result would be exactly the same. Fn1 tells the system to use the I and J values to define the arc centre – which is exactly what is wanted.

Fillet and Chamfer construction

A ***fillet*** is an arc which is tangential to two lines, a line and an arc, an arc and a line or two arcs. The direction of the fillet is chosen so that it maintains a continuity of direction from the first line or arc, around the fillet, and on to the following line or arc. Only the radius is specified – as a positive value in the Radius column. For a fillet between two lines or arcs to be possible, they must intersect. It is not possible to construct a fillet between two lines if they are parallel, nor is it possible to construct a fillet between a line and an arc if the line does not intersect the arc. Lines and arcs which are tangential are considered to intersect. In the diagram below example A and C have two elements which intersect, and so a fillet is possible between them. Example B shows an arc which is tangential to the line which precedes it, but because the direction is reversed at the point of tangency a fillet is also possible in this situation.

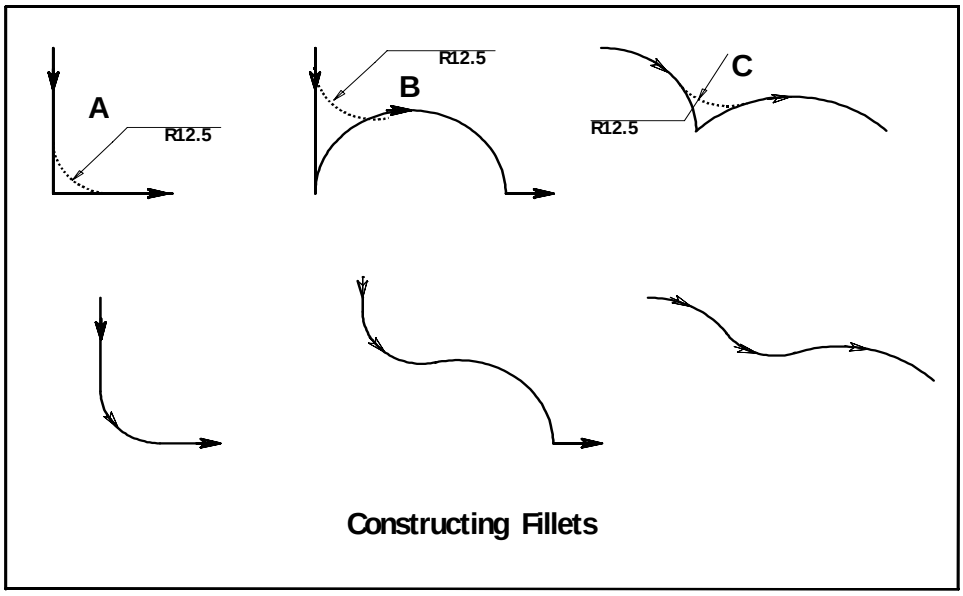


Figure 3

Program examples using fillets

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	35.000	0.000	0.000	0.000	0.000
2	L	0	0.000	0.000	0.000	0.000	0.000	12.500
3	L	0	25.000	0.000	0.000	0.000	0.000	0.000
1	P	0	0.000	35.000	0.000	0.000	0.000	0.000
2	L	0	0.000	0.000	0.000	0.000	0.000	12.500
3	CW	1	40.000	0.000	0.000	20.000	0.000	0.000
1	P	0	0.000	20.000	0.000	0.000	0.000	0.000
2	CW	1	20.000	0.000	0.000	0.000	-20.000	12.500
3	CW	0	60.000	0.000	0.000	0.000	0.000	28.280

Programs for figure 3

A chamfer is a short line segment inserted between two elements. Only the size of the chamfer is specified – by specifying a negative value in the radius column. A chamfer can be used anywhere that a fillet can be used.

To construct the chamfer first imagine a circle whose radius is the same as the chamfer size, drawn at the point where two geometry elements intersect. This circle will intersect both the two geometry elements. The chamfer is then the line segment joining the two intersection points.

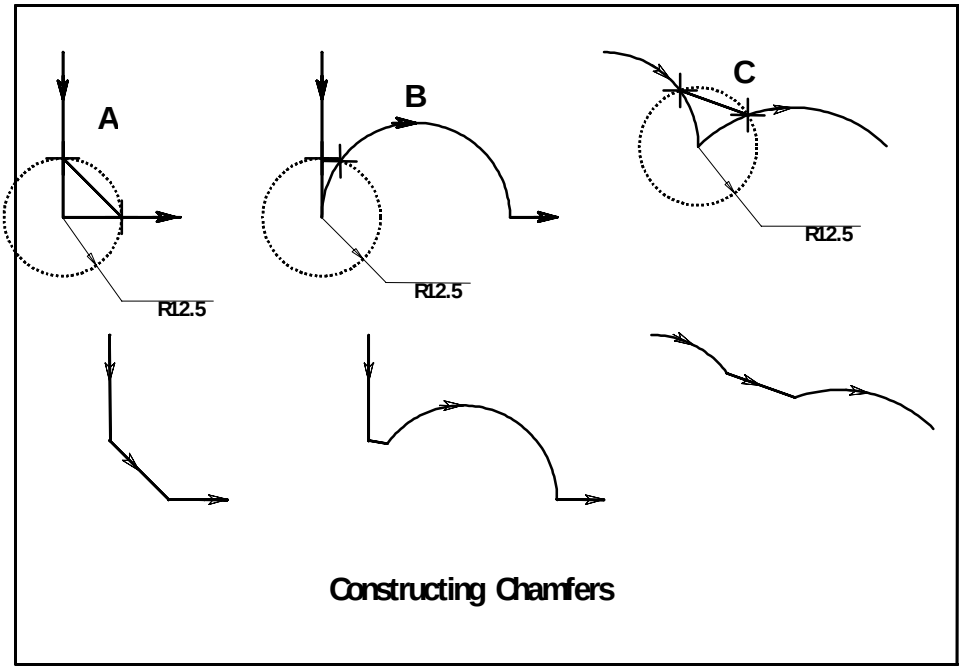


Figure 4

In each of the examples A, B and C, an imaginary circle has been placed at the intersection of the first and second geometry elements. The intersection points which are connected to form the chamfer are shown as +'s.

The resulting chamfers are shown in the lower half of the diagram

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	35.000	0.000	0.000	0.000	0.000
2	L	0	0.000	0.000	0.000	0.000	0.000	-12.500
3	L	0	25.000	0.000	0.000	0.000	0.000	0.000
1	P	0	0.000	35.000	0.000	0.000	0.000	0.000
2	L	0	0.000	0.000	0.000	0.000	0.000	-12.500
3	CW	1	40.000	0.000	0.000	20.000	0.000	0.000
1	P	0	0.000	20.000	0.000	0.000	0.000	0.000
2	CW	1	20.000	0.000	0.000	0.000	-20.000	-12.500
3	CW	0	60.000	0.000	0.000	0.000	0.000	28.280

Program segments for figure 4

Note that in each program segment the only difference – compared to the fillet example in Figure 3 – is the sign of the value in the Radius / Chamfer column. To define a chamfer it must be negative.

Complex Geometry

As stated earlier *complex geometry* occurs when it is not possible to simply read the end points of each line and arc segment, defining a contour or pocket boundary, directly from the component drawing. In this case the end points must be calculated using geometric functions.

The system has two groups of functions for dealing with *complex* geometry

1) ***The intersection functions***. These are used to find intersection points between two lines, a line and an arc, or between two arcs. There are some simple rules governing the use of the intersection functions :-
   a) The intersection functions are always used in pairs. The first specifies that the end point of the Line or Arc step is to be solved as an intersection, the second specifies that the start point of the Line or Arc step is to be solved as an intersection.
   b) The intersection point is common to the two Line / Arc steps.
   c) If one or both of the Steps involved in the intersection is an arc, then two solutions are possible. The desired solution is specified by the intersection Function number.
   d) Arcs involved in intersections **must** be defined using I and J parameters.
   e) It is possible to insert a Fillet or Chamfer between any two Steps involved in an intersection.
   f) The only Step that can appear in between a pair using intersection functions is an *auxiliary point* (q.v.) definition..
2) ***The Tangent functions***. These are used to find the point where a line or an arc is tangent to either the arc or line defined in the previous step or, the arc or line defined in the following step. As with fillets there are some simple rules to observe …
   a) A Line can be tangent to …
      i) The arc defined in the previous step. – The end point of the line must be known.
      ii) The arc defined in the next step. – The start point of the line must be known.
      iii) The arcs defined in both the previous and the next steps.
   b) An arc can be tangent to …
      i) The arc or line defined in the previous step, the end point of the new arc and its radius must be known.
      ii) The arc or line defined in the next step. The start point and radius of the new arc must be known.
      iii) The arc or line defined in both the previous and the next steps. Only the radius need be known. (If the arc or line defined in the previous and the next step intersect, the tangent arc can usually be replaced by a fillet).
      iv) The line defined in the previous step, and two points – one of which is the end point of the previous line, the other is the end point of the new arc. Only the direction of the arc needs be known. The system will calculate the centre point and radius.

Using the Intersection functions

Line intersections

In the example below, it is not possible to determine the end point of the line A-B simply by reading it from the drawing. It could be calculated using geometric functions, but it is much simpler to get to calculate the point using one of its ***Intersection functions***

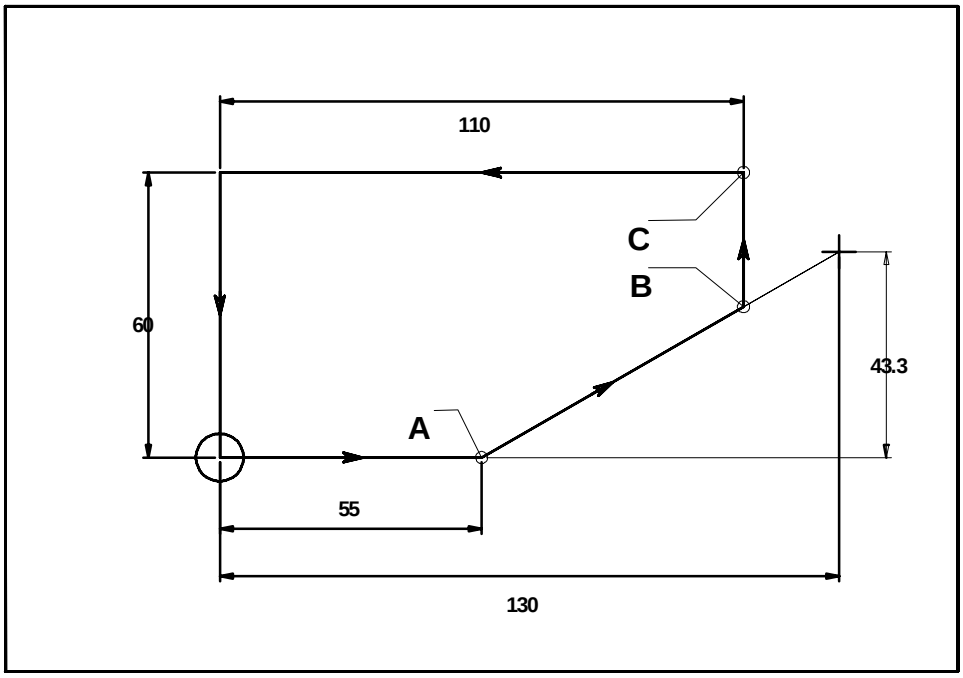


Figure 5

From the drawing, the end point of the line A-B is unknown (point B), but it is known that it passes through the point X130 Y43.3, and its start point is also known. What is needed is a function that defines a line starting at a known position, and passing through a known point. The end point of the line will be where it intersects the line B-C.

*Line **Fn2** (and **Fn3**) is used to define a line between two points. The line has a fixed start point i.e. the end point defined in the previous step, and ends at its intersection point with the line or arc defined in the next step. The values placed in the X and Y columns define a point on the line, which may or may not be the actual end-point.*

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	55.000	0.000	0.000	0.000	0.000	0.000
3	L	2	130.000	43.300	0.000	0.000	0.000	0.000

The 1st three steps used to program figure 5

Step 3 in this program (partially) defines the line A-B. Fn2 specifies that the line passes through the point X130 Y43.3, but that the line ends where it intersects the line about to be defined in step 4. This will be the line B-C.

The line B-C represents a slightly different problem, it's start point is not known. It is known that it ends at X110 Y60, and that its direction is in the Y+ direction or 90 degrees (from the X axis). It is also known that if the line A-B were removed from the drawing then the line B-C would be from X110 Y0 to X110 Y60.

What is needed is a function that allows a line to be defined which starts at an intersection point (with the line A-B) and ends at a known point, and has a specific direction. Line Fn4 and 5 are used to define lines which pass through a particular point at a known angle.

Line Fn4 (and Fn5) specifies a line which passes through the point defined in the X and Y columns. The value in the I column (the heading will change to "Angle") is the angle that the line makes with the X axis. The point defined in the XY columns will be either the **start** point of the line or the **end** point of the line, depending on the context. If the previous step contained an **intersection** function code – as in the example above – then the point is the **end** of the line, otherwise it is the **start** of the line.

Using Function 4 to complete the program …

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	55.000	0.000	0.000	0.000	0.000	0.000
3	L	2	130.000	43.300	0.000	0.000	0.000	0.000
4	L	4	110.000	60.000	0.000	90.000	0.000	0.000
5	L	0	0.000	60.000	0.000	0.000	0.000	0.000
E 6	L	0	0.000	0.000	0.000	0.000	0.000	0.000

The completed program for figure 5

What can be seen here in the complete program, is that intersection functions are always used in pairs Step 3 uses FN2 to define a line with a known start point – the XY coordinate in Step2, but whose end point is not known. Step 4 defines a line whose end point and direction are known but not the start point. Because Step 4 is the second intersection function of a pair its XY coordinate is the end of the line.

It was stated earlier that the line B-C is part of the line from X110 Y0 to X110 Y60. This provides an alternate way of defining the line using an **Auxiliary point** function.

A Point step defined with Fn1, defines an auxiliary point. This point is not necessarily on the contour or pocket being defined. It is used to define a temporary start point for the following step in situations where the actual start point is not known.
In the case of figure 5, an auxiliary point can be used to define a temporary start point for the line B-C. Here it is shown at Step 4 …

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	55.000	0.000	0.000	0.000	0.000	0.000
3	L	2	130.000	43.300	0.000	0.000	0.000	0.000
4	P	1	110.000	0.000	0.000	0.000	0.000	0.000
5	L	4	110.000	60.000	0.000	0.000	0.000	0.000
6	L	0	0.000	60.000	0.000	0.000	0.000	0.000
E 7	L	0	0.000	0.000	0.000	0.000	0.000	0.000

An alternate program for figure 5 using *Auxiliary* points.

Using Auxiliary points adds one line to the program. The auxiliary point definition is placed between the two lines using intersection functions. Step 4 now provides a temporary start point for the line defined in step 5.

Auxiliary points provide a convenient way to specify a temporary start point for the following Line or Arc step. If the following step is a line, it should be defined using Fn4 or Fn5 otherwise the auxiliary point will be discarded.

In figure 5, the line A-B was defined using a temporary end-point. Figure 6 shown below defines the line by specifying the angle it makes with the X axis.

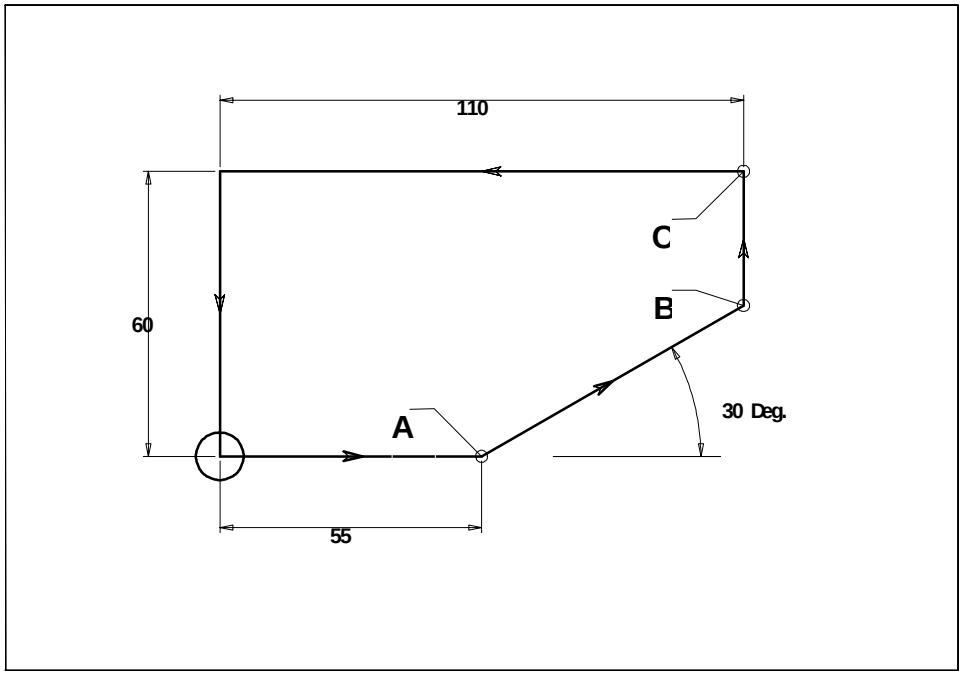


Figure 6

An alternate method of defining the line A-B is now required, the start point and angle it makes with the X axis - 30 degrees are known, but the end point is not. It was shown earlier that Line Fn4 (and Fn5) can be used to define a line passing throgh a point at a fixed angle. Figure 6 can be programmed using Fn4 to define both line A-B and line B-C.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	55.000	0.000	0.000	0.000	0.000	0.000
3	L	4	55.000	0.000	0.000	30.000	0.000	0.000
4	L	4	110.000	60.000	0.000	90.000	0.000	0.000
5	L	0	0.000	60.000	0.000	0.000	0.000	0.000
E 6	L	0	0.000	0.000	0.000	0.000	0.000	0.000

The completed program for figure 6

Remember that FN4 allows defines a line passing through a point at a known angle, and that the point defined in the XY columns can be either the start point of the line ***or*** the end point of the line.

When FN4 is used to define two consecutive lines as in Steps 3 and 4, the XY coordinate of the first line is its start point, and the XY coordinate of the second line is its end point.

A summary of Line intersection functions

Line Function 2 (and Function 3) is used to define a line which passes through two known points. The first point is the end point of the previous step. The second point is a temporary end point.

Line Function 4 (and Function 5) is used to define a line which passes through a known point at a given angle. The known point may be either the start point of the line, or it may be the end point of the line. When defining lines using Function 4 or 5, the I/Angle column contains angle the line makes with the X axis.

Point Function1 defines an *auxiliary* point. This point may be used to provide a temporary start point for the line which follows it. The line will be programmed using Function 4 or 5 ; the angle of the line will be defined by its temporary start point and its programmed end point (and will override the value programmed in the I / Angle column).

Where choices exist

It has been stated earlier that Line Function 2 and Line Function 3 do the same thing. I.e. They define a line passing through a known start point and a temporary end point,so why do we have two function numbers ? Because intersection involving arcs have two solutions .

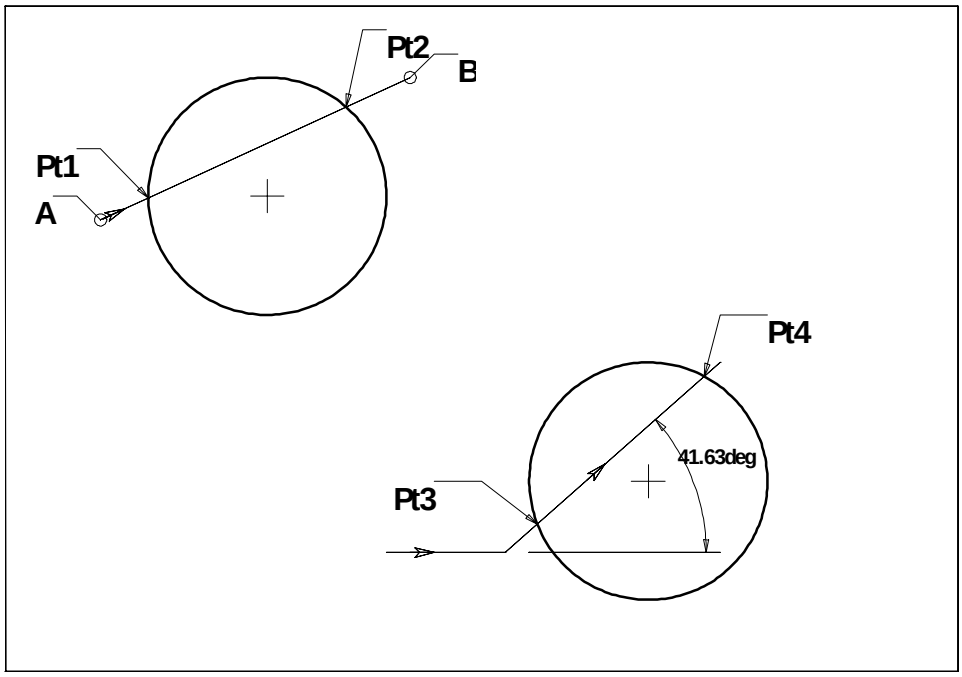


Figure 7

Remember …

Line Function 2 (and Function 3) is used to define a line which passes through two known points – A-B in the diagram above. The first point is the end point of the previous step. The second point is a temporary end point. **There are two line functions FN2 and FN3 so that if the line intersects an arc, it is possible to choose one intersection point in preference to the other, one of the functions must choose the point Pt1, and the othe must choose Pt2.**

Similarly, Line Function 4 (and Function 5) is used to define a line which passes through a fixed point at a given angle. Function 4 must select either Pt3 or Pt4, and Function 5 must select the other.

Clearly there must be a method of predicting which of the intersection functions will select the correct point.

Assume that the line A-B is to be programmed using either FN2 or FN3, and that it must end at either Pt1 or Pt2 which function will select Pt1 and which function will select Pt3 ?

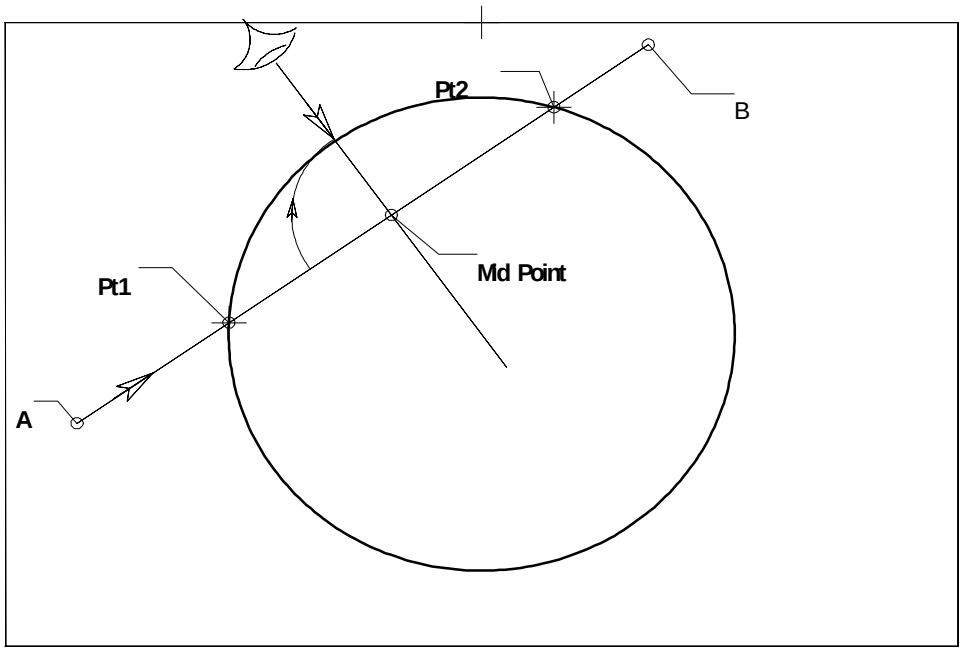


Figure 8

The Line A-B starts at the point A and has the direction indicated by the arrow, towards point B. It intersects the circle at two points Pt1 and Pt2. Take the line segment Pt1-Pt2 and rotate it through 90 degrees clockwise, this produces the dotted line. Sight along the rotated line as shown in Figure 8. ***FN2 will select the intersection point on the left hand side of the line and Fn3 will select the point on the right hand side of this line.***

In other words, if the line is to end at the point Pt1 it must be programmed using FN3 and if it is to end at the point **B** then FN2 must be used.

The same rule applies if the Line is defined as a line at an angle using either FN4 or FN5. In this case to select point Pt1 use FN5, and to select point Pt2 use function FN4.

Line and arc intersections

Using the rule described above to differentiate between two possible solutions the component show in Figure 9 should present no difficulty.

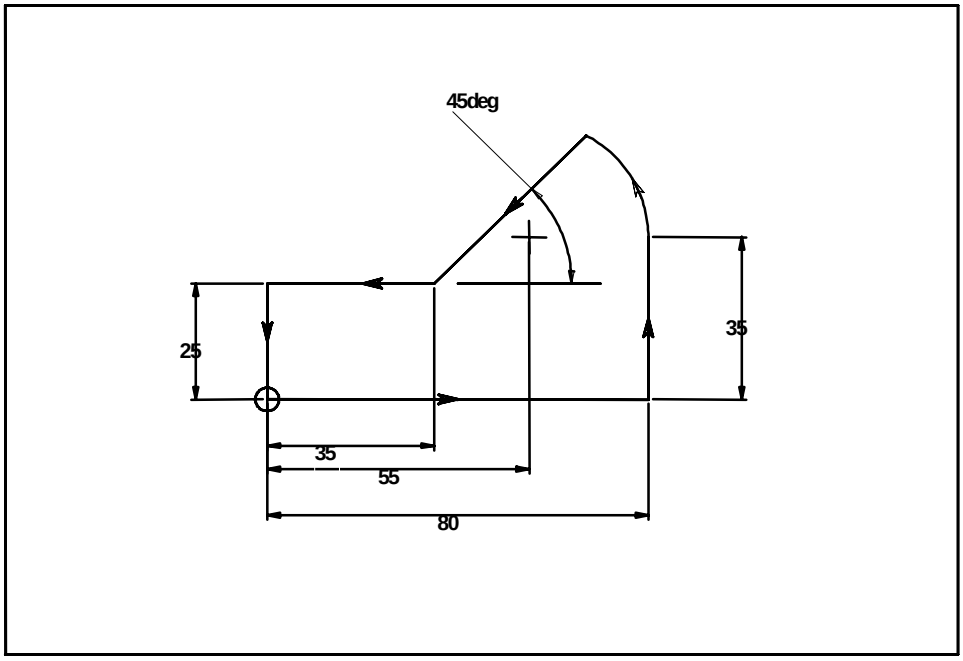


Figure 9

Starting from the point X0 Y0 and following the directions of the arrows the program starts of like this …

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	80.000	0.000	0.000	0.000	0.000	0.000
3	L	0	80.000	35.000	0.000	0.000	0.000	0.000
4	CCW							

Figure 9, the first Steps

The first three steps can be programmed easily using FN0. Step 4 is the CCW arc.

The arc's centre point can be easily read from the drawing, and its start point and direction are known. What is needed is an Arc function that behaves like the Line intersection functions.

Arc Fn2 (and Fn3) *Specifies an arc that starts at a fixed point i.e. the end point of the previous step, and ends at its intersection with the following line or arc. By implication the following line or arc starts at the same intersection point, and ends at its programmed XY position. The arc must be programmed using I and J (incremental from the start point) to define the centre point, and is initially assumed to define 360 degree arcs.*

The rule[3] for determining whether to use FN2 or FN3 must now be applied …

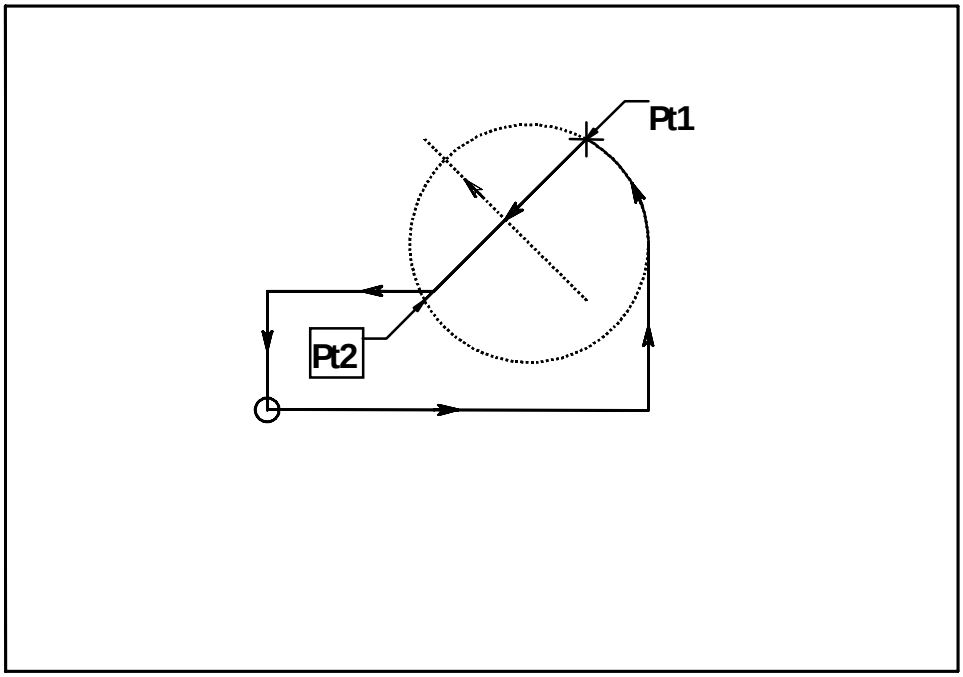


Figure 10

In order to apply the Left/Right hand rule the line segment Pt1-Pt2 is rotated through 90 degrees clockwise to produce the dotted line shown in figure 10. Sighting along this line, it can be seen that Pt1 – the desired intersection point is to the right of the dotted line, therfor FN3 will give the correct intersection.

Step 4 of the program can now be entered …

4	CCW	3	80.000	35.000	0.000	-25.000	0.000	0.000
5								

[3] Where two intersections exist, FN2 will select the Left hand intersection and FN3 will select the right hand intersection.

In step 4, the end point of the arc has been programmed at X80 Y35 – which happens to be the start point. In fact any convenient point on the arc could have been used as this is a temporary end-point, it will be replaced by the intersection point Pt1.

Step 5 is simply a line at an angle of 225 (or –135) degrees measured positive clockwise from the X axis. Line Function 4 (or Function 5) can be used to program this line.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	80.000	0.000	0.000	0.000	0.000	0.000
3	L	0	80.000	35.000	0.000	0.000	0.000	0.000
4	CCW	3	80.000	35.000	0.000	-25.000	0.000	0.000
5	L	4	55.000	25.000	0.000	225.000	0.000	0.000
6	L	0	0.000	25.000	0.000	0.000	0.000	0.000
E 7	L	0	0.000	0.000	0.000	0.000	0.000	0.000

The completed program for Figure 9

Steps 6,7 complete the program, and are simple Line steps requiring no special functions.

Before moving on to the next example it is important to see how reversing the direction of this shape (i.e. Figure 9) from a generally CCW shape to a generally CW shape affects the programming, particularly the line / arc intersection point.

The program starts simply enough at X0 Y0 and follows the vertical line to X0 Y25

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	0.000	25.000	0.000	0.000	0.000	0.000
3	L	0	55.000	25.000	0.000	0.000	0.000	0.000
4	L							

Up to Step 4, Fn0 can be used throughout, but now Step 4 must be programmed as a line at an angle of 45 degrees through the point X55 Y25, and it must also take responsibility for selecting the correct intersection with the arc (Pt1 or Pt2 in Figure 10). Because the direction of the shape has bee reversed, the direction of the dotted line show in Figure 10 must also be reversed. The desired point Pt1 is now on the LeftHandSide of the dotted line. This implies that FN4 must be used to program the line.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	0.000	25.000	0.000	0.000	0.000	0.000
3	L	0	55.000	25.000	0.000	0.000	0.000	0.000
4	L	4	55.000	25.000	0.000	45.000	0.000	0.000
5	CW							

The next step is the arc, which is now CW, but there is a problem, it cannot be programmed as a simple arc (FN0 or FN1) because its start point is unknown. Obviously it is the point Pt1, but Pt1's coordinates are not known. The arc's centre point is known as is its end point, and direction. There is an arc function FN10 which copes with this problem ..

Arc Fn10 *can be used to program an arc whose start point is unknown (because it is an intersection or tangent point with the previous step). The arc's centre point is programmed using* ***I*** *and* ***J*** *parameters measured from the* ***end point*** *of the arc.*
So the program can be completed using FN10 to define the arc, and simple Line functions to complete the shape.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	0.000	25.000	0.000	0.000	0.000	0.000
3	L	0	55.000	25.000	0.000	0.000	0.000	0.000
4	L	4	55.000	25.000	0.000	45.000	0.000	0.000
5	CW	10	80.000	35.000	0.000	-25.000	0.000	0.000
6	L	0	80.000	0.000	0.000	0.000	0.000	0.000
E 7	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Arcs with more than one line intersecting

Figure 11 poses a slightly more complex problem, there are two pairs of intersections involving the circle. How many different shapes can be defined using the rectangle and the circle ?.

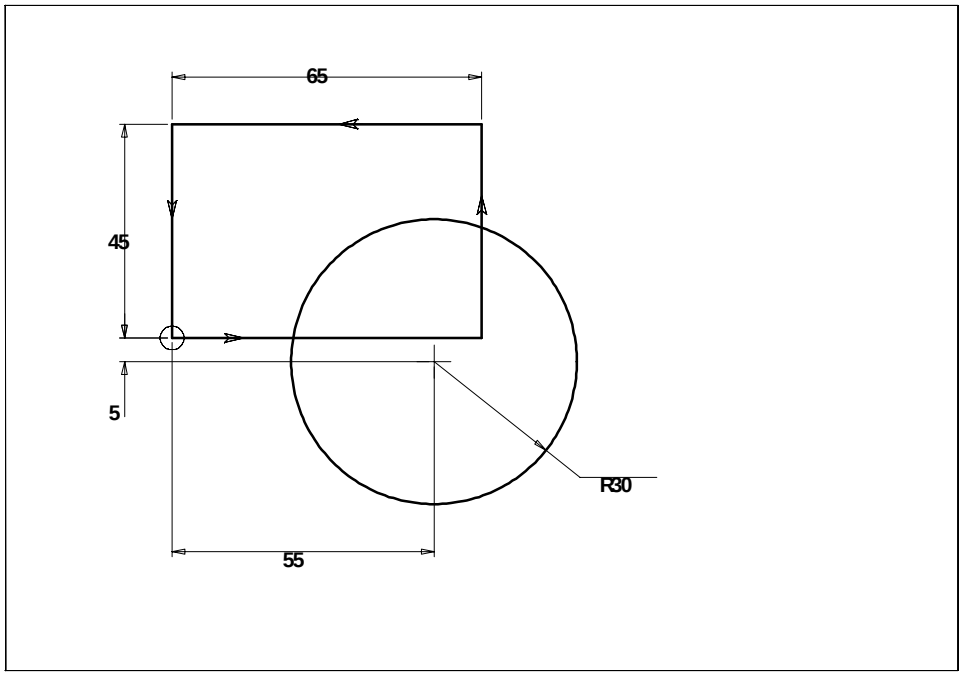


Figure 11

Two shapes ? Four ? Six ?.

There are eight (2 intersections with the horizontal line * 2 intersections with the vertical line * 2 directions = 8), but not all of them are useful shapes.

Here are some of them …

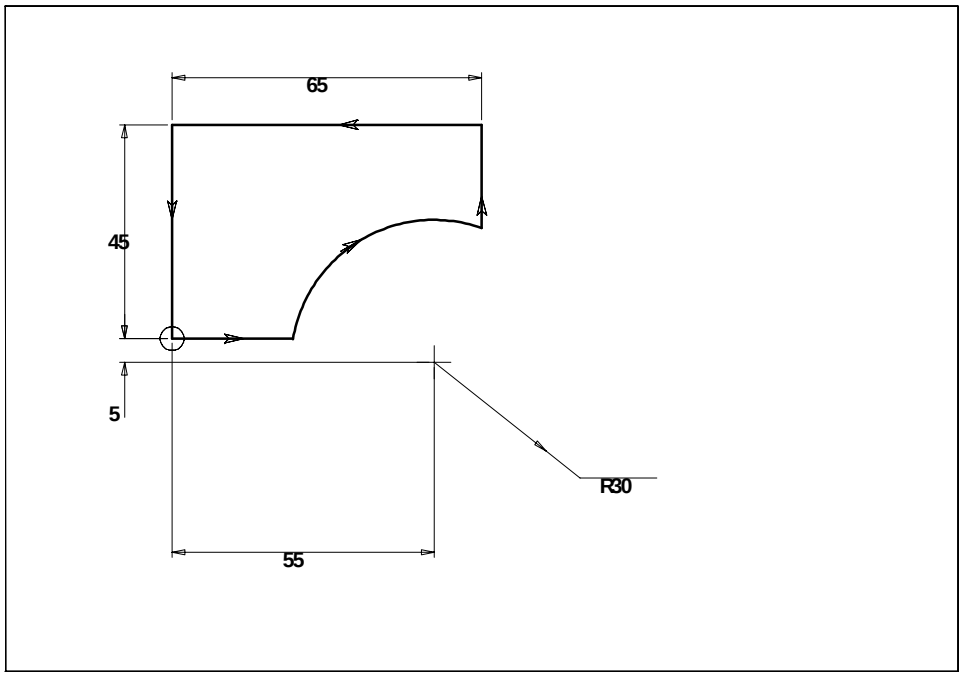


Figure 12

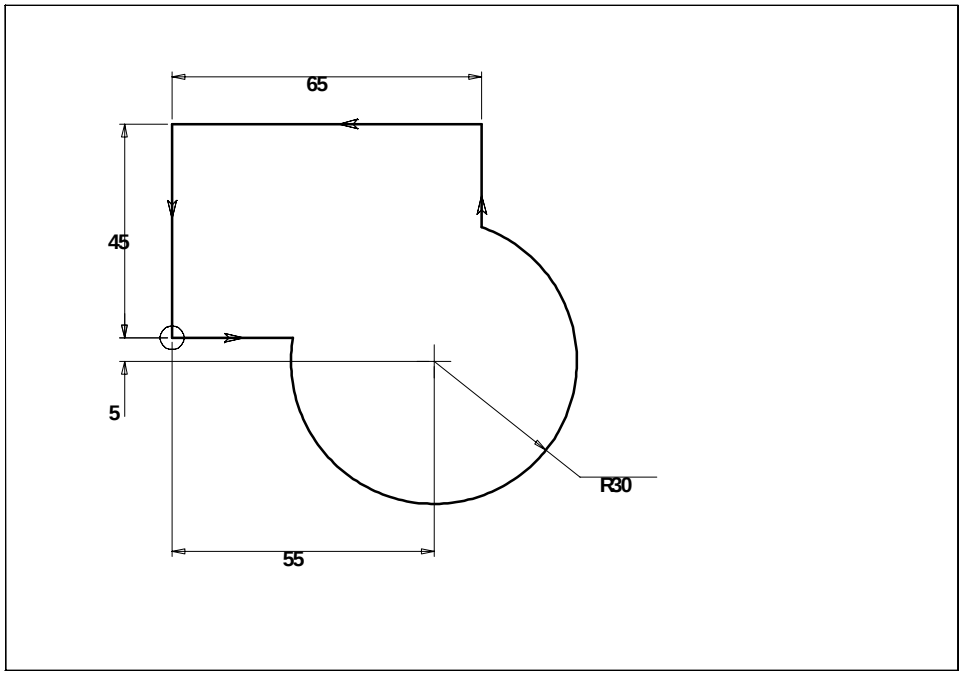


Figure 13

The only difference between the two shapes shown in Figure 12 and Figure 13 is the direction of the arc. In figure 12 the arc is CW and in figure 13 the arc is CCW.

Now look at figure 14 and Figure 15. How do these shapes differ from figure 13 ? What causes these shapes to be different and how are these different shapes defined.?

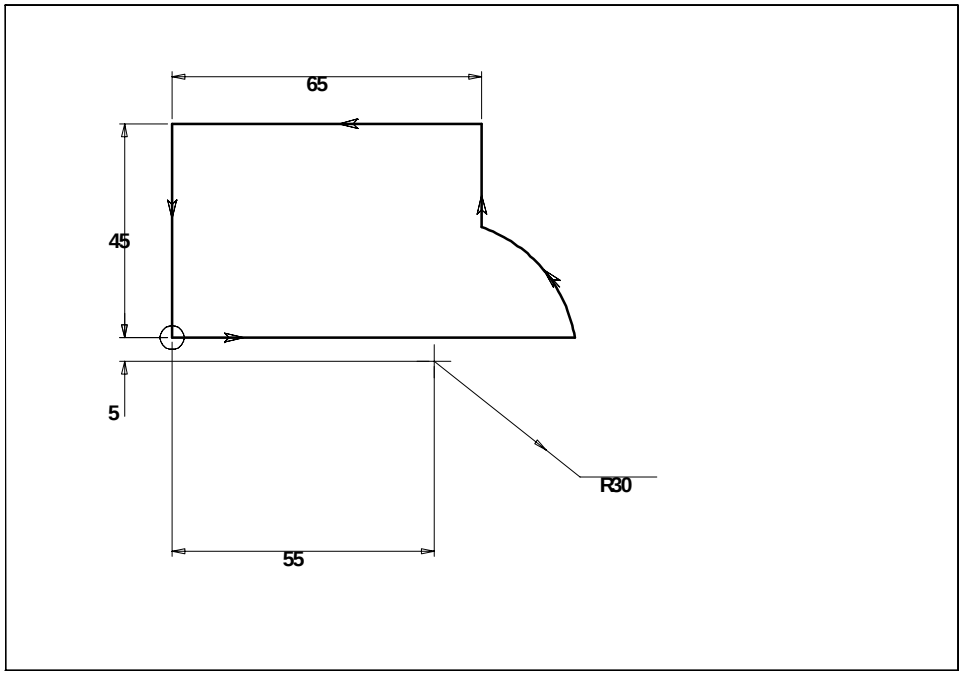


Figure 14

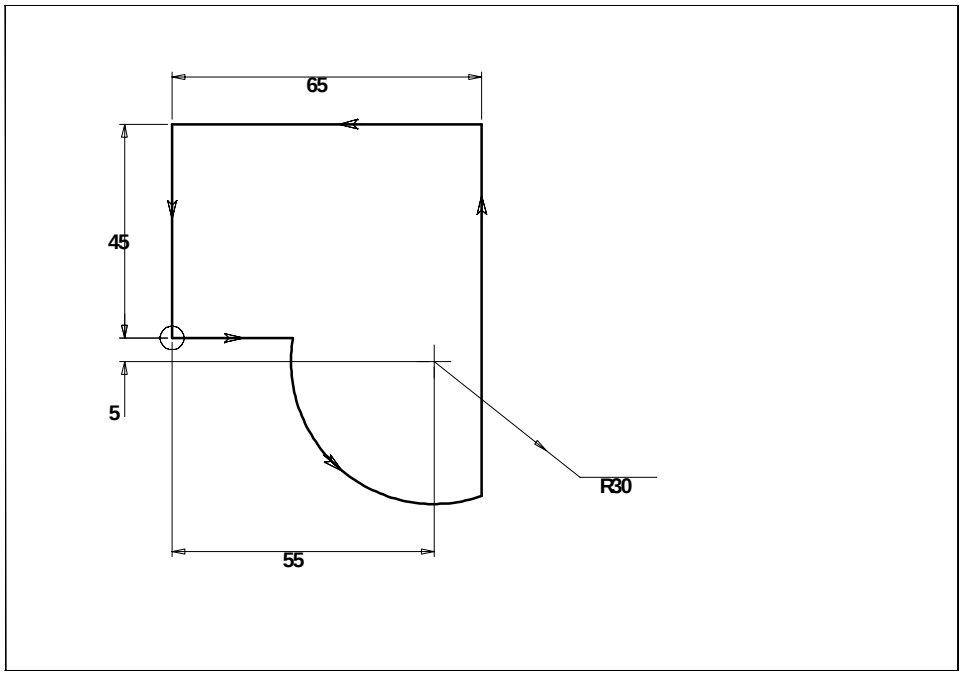


Figure 15

All these shapes (Figures 12, 13, 14 and 15) are possible because when a line and a circle intersect each other there are ***two intersections points***.

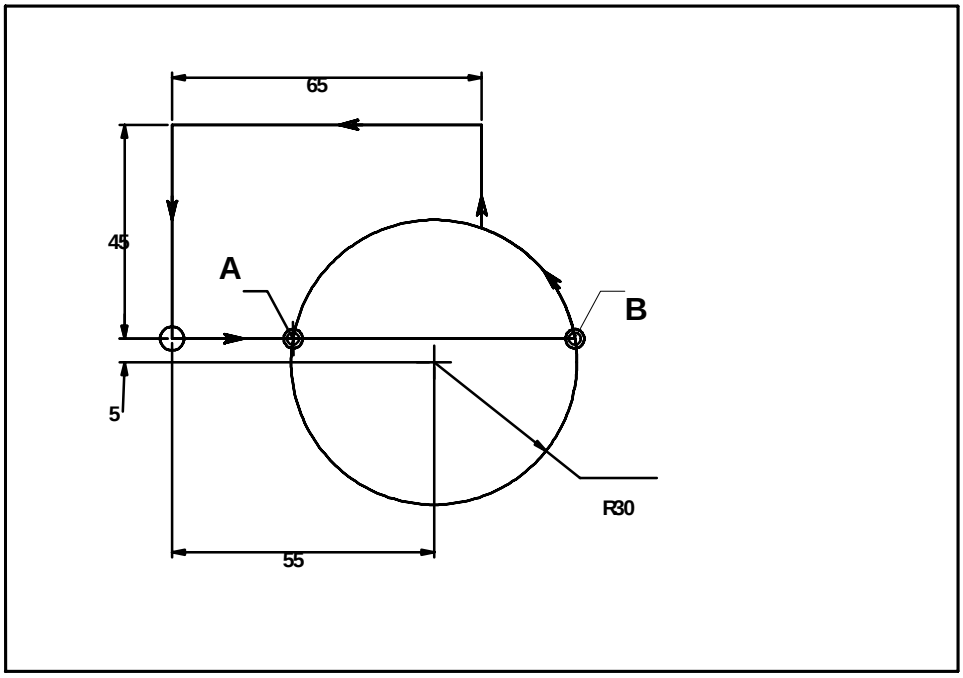


Figure 16

When considering the first line from X0 Y0 to X65 Y0 it can be programmed using Function 2, or Function 3 because they are both functions for defining a line through two points. The program for Figure 16 will begin like this …

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	2	65.000	0.000	0.000	0.000	0.000	0.000

Or this …

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	3	65.000	0.000	0.000	0.000	0.000	0.000

The first and second program differ only by the function number in step 2. The Left / Right hand rule described earlier defines which is the correct function to select.

Going back to Figure 16, it is now time to define the arc. What is known about it ?

- It starts at the point where the line X0 Y0 – X65 Y0 intersects the arc, but the coordinate of that point is not known
- It ends where the line X65 Y0 – X65 Y45 intersects the arc, but again the coordinate of that point is also unknown.
- The arc's direction is known – it can easily be inferred from the drawing.
- The arc's centre point is known – it is specified in each of the figures 11 thro15
- The arc's radius is known, again it is specified in the drawings.

The arc cannot be programmed using CW Fn0 or CCW Fn0 because neither of the end points is known, and so the arc must be defined using I and J values.

I and J define the centre point of an arc relative to its start point, but the coordinate of the start point is not known, a *temporary start point* must be used. An ***auxiliary point*** could be used to act as a temporary start point. This would then allow the arc to be programmed using I and J values to define the centre. The programmed end point would also have to be a temporary point.

Given that the arc's centre point and radius are known it is very easy to calculate one of the quadrant points on the arc. E.g.

X = X centre +/- Radius, Y = Y centre
Or,

X = X centre, Y = Y centre +/- Radius
This provides four points to choose from to act as a temporary start / end point for the arc. It doesn't matter which is selected if it is assumed for the time being, that the arc is a full circle.

Remember from the discussion about Figure 9 …

Arc Function 2 (and Function 3) defines an arc which starts at a known point i.e. the end of the previous step, and ends at its intersection with line or arc defined in the next step. There are two functions because there is always two possible solutions. Tthe intersection chosen to act as the arc end point uses the Left / Right hand rule.
The system understands that although the start point of the arc is unknown, as soon as the centre point and radius are defined, the start point can be calculated. Then the arc complies with the rule above and all that remains to be solved is the end point.

Adding the arc step to the program gives …

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	2	65.000	0.000	0.000	0.000	0.000	0.000
3	CW	3	55.000	25.000	0.000	0.000	-30.000	0.000

X55 Y25 has been programmed as the temporary start / end point for the arc. This is the quadrant point at the 12 o'clock position. From this temporary start point, the arc centre can be defined using I0 J-30.

The right hand vertical line, i.e. the one that ends at X65 Y45 can now be programmed. This is the same problem seen in Figure 5. Function 4 can be used to define this line.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	3	65.000	0.000	0.000	0.000	0.000	0.000
3	CW	2	55.000	25.000	0.000	0.000	-30.000	0.000
4	L	4	65.000	45.000	0.000	90.000	0.000	0.000

It does not matter which Function is used 4 or 5, because the intersection point (from the arc to this line) is selected by the Function associated with the arc in step 3.

The program can be completed using Line steps defined with FN0 / FN1 …

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	3	65.000	0.000	0.000	0.000	0.000	0.000
3	CW	2	55.000	25.000	0.000	0.000	-30.000	0.000
4	L	4	65.000	45.000	0.000	90.000	0.000	0.000
5	L	0	0.000	45.000	0.000	0.000	0.000	0.000
E 6	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Which figure does this program[4] relate to 12, 13, 14, or 15. ?

[4] The shape defined by this program is the one shown in Figure 12.

Intersection of two arcs

Figure 16A shows a contour with two intersecting arcs …

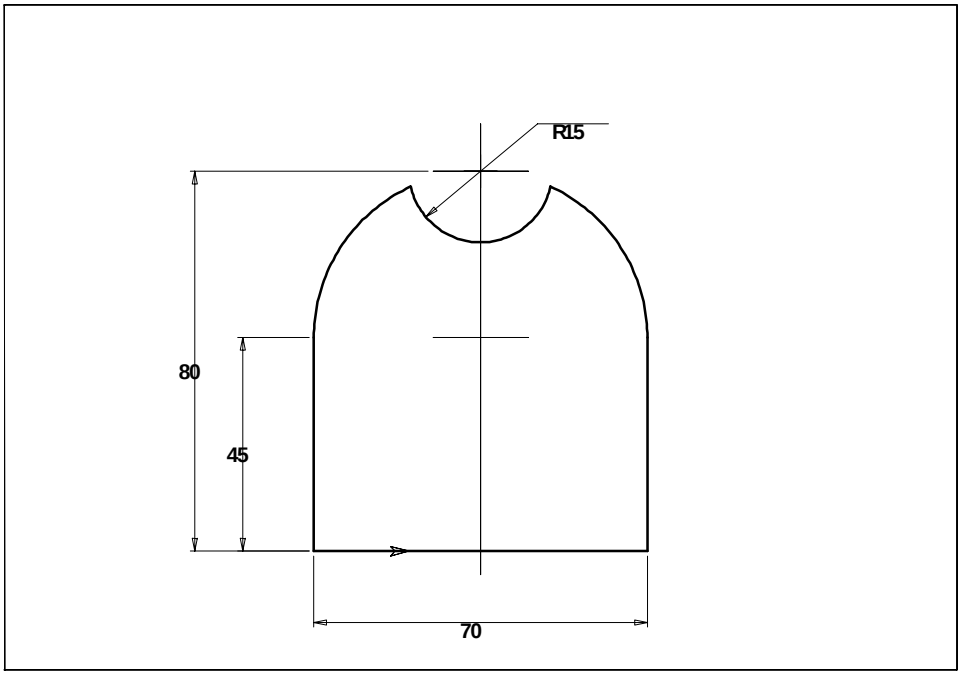


Figure 16A

The main problem that needs to be solved is that the start and end points of the R15 arc are both unknown, and there is no fucntion for defining an arc in which both end-points are to be solved as intersections.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	70.000	0.000	0.000	0.000	0.000	0.000
3	L	0	70.000	45.000	0.000	0.000	0.000	0.000
4	CCW	3	35.000	80.000	0.000	35.000	0.000	0.000
5	CW	10	35.000	65.000	0.000	0.000	15.000	0.000
6	CW	3	20.000	80.000	0.000	0.000	15.000	0.000
7	CCW	10	0.000	45.000	0.000	35.000	0.000	0.000
E 8	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Figure 16A

Steps 1- 3 are simple geometry with no special functions. Step 4 Uses the Arc intersection function FN3 to indicate that it will end at its intersection with the arc in step 5. The X and Y coordinate is a temporary end-point at the 12 o'clock position.

Steps 5 and 6 split the 15mm radius arc into two sections at the 6o'clock position. This eliminates the problem of finding two intersections on the arc, by splitting into two pieces, and finding one intersection on each piece using Fn3. The first half of the arc uses FN10 – because its start point is unknown – while the second half uses FN3 to find its intersection with the left hand part of the R35 arc.

Steps 7 defines the seconds – left hand – R35 arc using FN10 so that its I and J values can be programmed relative to the end -point.

Some more Fillets and Chamfers

In using the intersection functions, that is Line functions 2, 3, 4 and 4, and Arc functions 2, 3 and 10, no use has been made of the Radius / Chamfer column. Whenever arcs have been defined using intersection functions, it has been necessary to use I and J to define the arc centre point.

This provides the opportunity to use the Radius / Chamfer column to insert fillets between line / line intersections, between line / arc intersections and between arc / arc intersections.

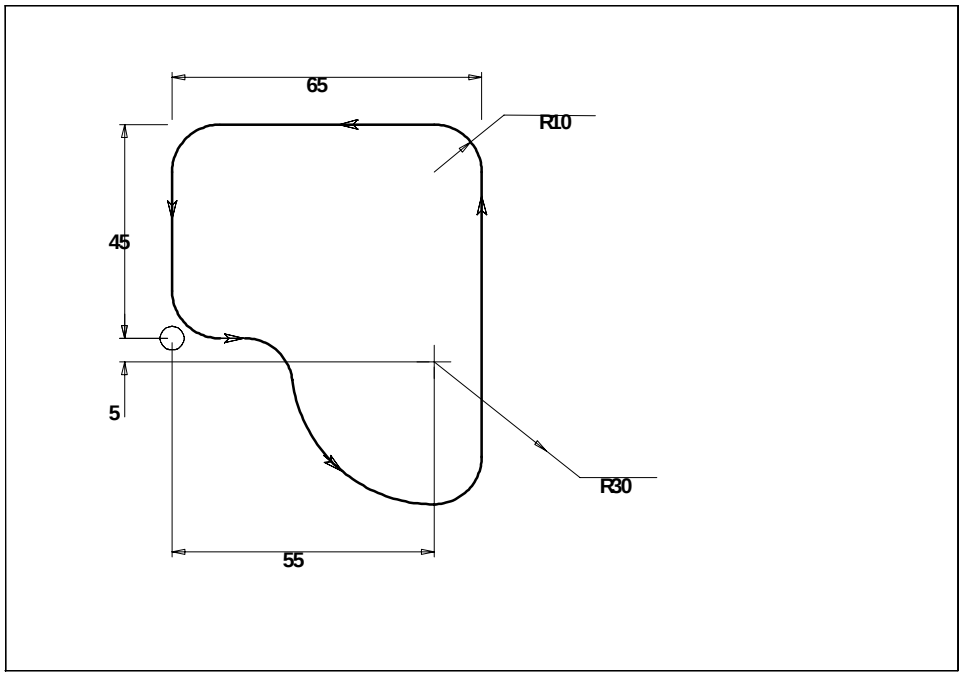


Figure 17

Figure 17 is the same shape as Figure 15 but with the addition of a 10mm radius fillet at every corner.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	3	65.000	0.000	0.000	0.000	0.000	10.000
3	CCW	3	55.000	25.000	0.000	0.000	-30.000	10.000
4	L	4	65.000	45.000	0.000	90.000	0.000	10.000
5	L	0	0.000	45.000	0.000	0.000	0.000	10.000
E 6	L	0	0.000	0.000	0.000	0.000	0.000	10.000

Setting the values in the Radius column to zero to produce a program for Figure 15, or as shown produces a program for Figure 17.

The Fillet attached to Step 6, is constructed between the line in Step 6 and the Line in Step 2. The Start point defined in Step 1 is no longer on the shape.

What has been achieved here is the simple straightforward programming of a shape in which not one of the constituent line or arc segment end points is known explicitly.

Using the Tangent Functions

In many ways the ***tangent*** functions are easier to use than the intersection functions, especially as system uses the direction of each geometry element to determine which tangent should be selected. There is always a choice of tangents as at least one of the steps involved will be an arc. There is always at least one arc involved when using the tangent functions.

Before describing the tangent functions it is important to understand the concept of *fixed* and *floating* lines arcs and points.

Fixed point
Is one which is known explicitly. Typically and start or end point of a line or arc.

Floating point
Is one which is defined as an intersection or a tangent point.

Fixed arc
Is one in which the centre point, radius and direction are all known.

Floating arc
Is one in which the position of the centre point is not known. The radius and direction must be known however.

Fixed line
A line is fixed if two points on the line are both known, or one point and the angle the line makes with the X axis are both known.

Floating line
A line is floating if only one point on it is known, or, if the line is tangent to two arcs, nothing else is known about it.

The task befalling the tangent functions is to define one *floating* line or arc, with respect to two *fixed* objects. This can be summarised as

1) A tangent function can be used to define a ***floating line*** relative to
   a) a fixed point and a fixed arc, or
   b) two fixed arcs.
2) A tangent function can be used to define a ***floating arc*** relative to
   a) A fixed point and a fixed arc **or** line
   b) Two fixed arcs.
   c) A fixed line and two fixed points – one of which is the end point of the fixed line.

NOTE: there is no tangent function to create a floating arc tangent to two fixed lines, or any two fixed objects (line or arcs) which intersect. This is a ***fillet***.

In the case of 2c above only the direction of the floating arc need be known.

Defining Lines tangent to a fixed arc

The simplest tangent functions are used to define a line which starts at a known point, and ends at a tangent point on the following arc, ***or*** which starts at a tangent point on the preceding arc and ends at a known point. This gives 3 tangent functions for line steps …

Line Function 6 *Specifies a line* ***starting*** *at a fixed point i.e. the end point of the previous step, and ending at its point of tangency with the following CW or CCW step. The tangent point is chosen to preserve continuity of direction from the line to the arc.*

Line Function 7 *Specifies a line* ***ending*** *at a fixed point entered in the X and Y columns, and starting at its point of tangency with the previous CW or CCW step. The tangent point is chosen to preserve continuity of direction from the arc to the line.*

Line Function 8 *Specifies a line tangent to* ***both*** *the preceding and the following arc steps. The points of tangency are chosen to preserve continuity of direction from the first arc, via the line to the second arc. XYZIJR are all ignored*

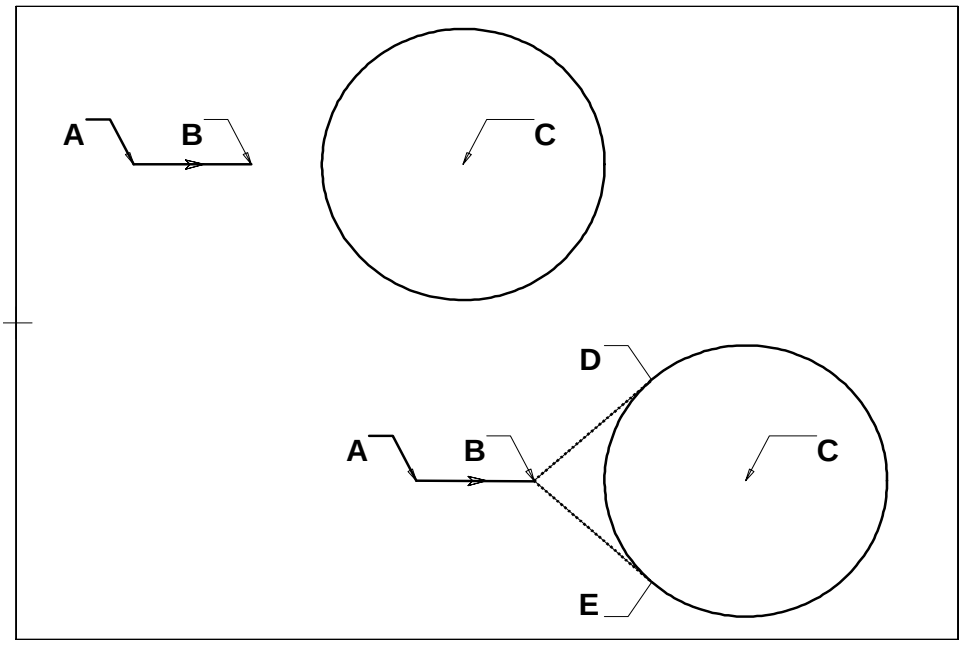


Figure 18

In the upper part of Figure 18 there is a line segment A-B. It is desired to define a line from the point B which is tangent to the circle centred at C.

As can be seen in the lower part of Figure 18, there are two lines which satisfy this requirement, B-D and B-E. Syetem has only a single function, Line Fn6 for specifying a line starting at a fixed point – B, and ending at a tangent point on the next arc. System uses the direction of the arc to determine which of the two tangents will be chosen …

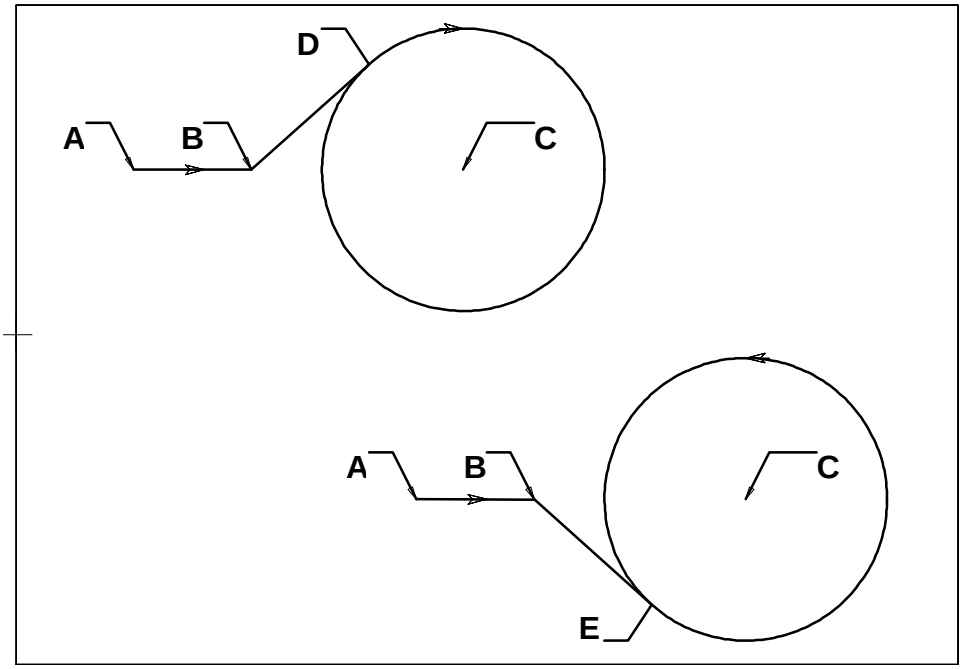


Figure 19

The circle in the upper part of Figure 19 has CW direction, therefore system selects the line B-D as the tangent. In the lower part of Figure 19, the circle has CCW direction and so system selects the line B-E as the tangent. Using this convention of maintaining continuity of direction simplifies the programming of tangent lines.

It is not possible to describe Tangent line functions without reviewing the arc function, FN10.

Arc Function 10 *Specifies an arc whose I and J values are relative to its end point. I.e. The arc start point is not known, typically because the previous step specified an intersection or tangent function.*

Two simple rules apply when programming Lines tangent to arcs

1. If the arc appears before the line, then it can be programmed as a simple arc using arc FN0 or arc FN1. The end point programmed in the X & Y columns is a temporary end-point and can be any convenient point on the arc, including the start point. The following line will then be programmed using Line FN7
2. If the arc appears after the line, then it cannot be programmed as a simple arc, FN10 must be used instead. This allows the I and J values to be programmed from the arc end point which is defined in the X & Y columns. The preceding line will have been programmed using Line FN6 to indicate that its end point and direction are unknown.

Tangent programming examples

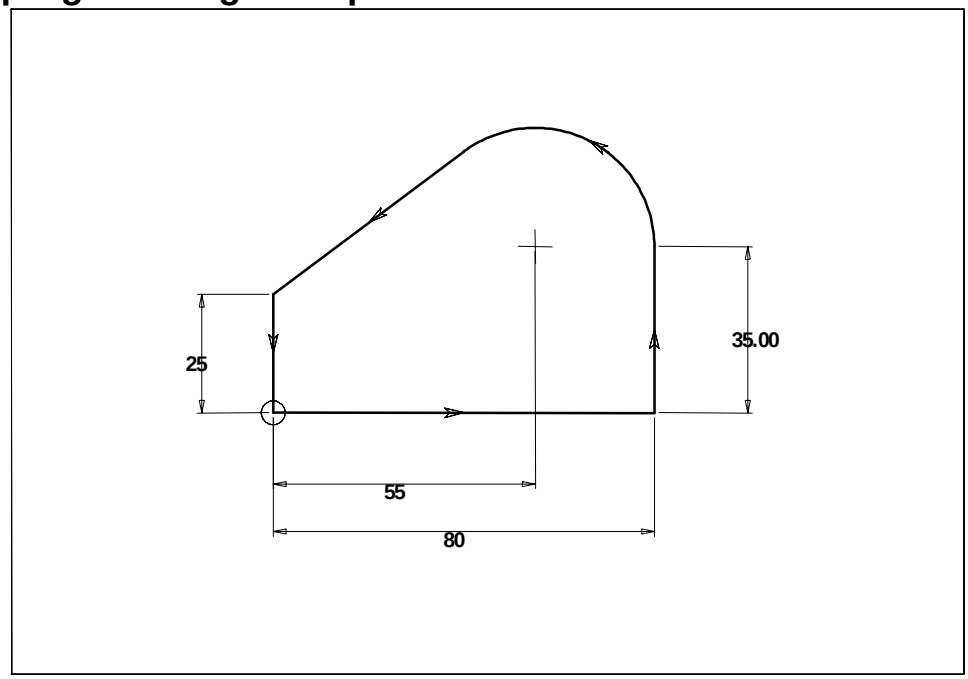


Figure 20

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	80.000	0.000	0.000	0.000	0.000	0.000
3	L	0	80.000	35.000	0.000	0.000	0.000	0.000
4	CCW	1	30.000	35.000	0.000	-25.000	0.000	0.000
5	L	7	0.000	25.000	0.000	0.000	0.000	0.000
E 6	L	0	0.000	0.000	0.000	0.000	0.000	0.000

The program for Figure 20

In this programming example the first three steps are straightforward.

When defining the arc the same trick is used as if it were to end at an intersection, a temporary end point is defined. It can be ***any*** convenient point on the arc. In this case a point at 9 o'clock on the arc was used. – *temporarily defining a semi-circle.* I and J values are easily calculated from the drawing data, I = X55 – X80 = -25 and J = 0. No special intersection or tangent function is required to define this arc. Function 1 is used simply to specify that the arc is programmed using I and J coordinates.

Step 5 uses ***Line FN7*** to define a line which starts at it tangent point with the arc in step 4. The X an Y columns contain the Line end point. The program is complete with a simple line back to X0 Y0.

If the direction of the contour is reversed, then the program changes

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	0.000	25.000	0.000	0.000	0.000	0.000
3	L	6	0.000	25.000	0.000	0.000	0.000	0.000
4	CW	10	80.000	35.000	0.000	-25.000	0.000	0.000
5	L	0	80.000	0.000	0.000	0.000	0.000	0.000
E 6	L	0	0.000	0.000	0.000	0.000	0.000	0.000

The program for Figure 20, reversed

Reversing the direction of the contour now means that the pair of steps involved in the tangent point calculation is steps 3 and 4.

In Step 3, Function 6 is used to specify that the line will end at the tangent point with the arc in Step 4. The X and Y values entered for Step 3 are the ***start*** point of the line. The end point will be calculated by the system.

In Step 4, the arc has changed direction and is now CW, the function number has changed to ***FN10 to indicate that the arc start point is unknown*** i.e. it *starts* at the tangent point with the line defined in Step 3. The X and Y columns contain the true end point of the arc. As with the previous examples using intersection functions, because the start point of the arc is not known, I and J are programmed using a *temporary start point*, which for convenience is assumed to be the point defined in the X & Y columns, i.e. the arc is treated as if it were a circle.

The remainder of the shape is programmed as simple line functions.

Using auxiliary points to define the arc start

There is an alternate method of defining the arc in step 4, and that is to place an auxiliary point definition in between steps 3 and 4. This auxiliary point can the act as a temporary start point for the arc, and the arc's I and J values can be programmed relative to the auxiliary point.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	0.000	25.000	0.000	0.000	0.000	0.000
3	L	6	0.000	25.000	0.000	0.000	0.000	0.000
4	P	1	55.000	60.000	0.000	0.000	0.000	0.000
5	CW	10	80.000	35.000	0.000	0.000	-25.000	0.000
6	L	0	80.000	0.000	0.000	0.000	0.000	0.000
E 7	L	0	0.000	0.000	0.000	0.000	0.000	0.000

An alternate method of programming Figure 17, reversed, using an auxiliary point

Step 4 defines an *auxiliary point* at X55 Y60, which is at 12 o'clock on the arc. This point acts as a temporary start point for the arc in Step 5. Now the I and J values used in Step 5 must relate to the start point defined in the previous block (which is the normal way of programming arcs using I and J).

Both this and the preceding method – using the end point of the arc as a temporary start point – will produce the same results.

A Line tangent to two arcs

Figure 21 shows a contour in which the two dotted circles are to be joined by a tangent line.

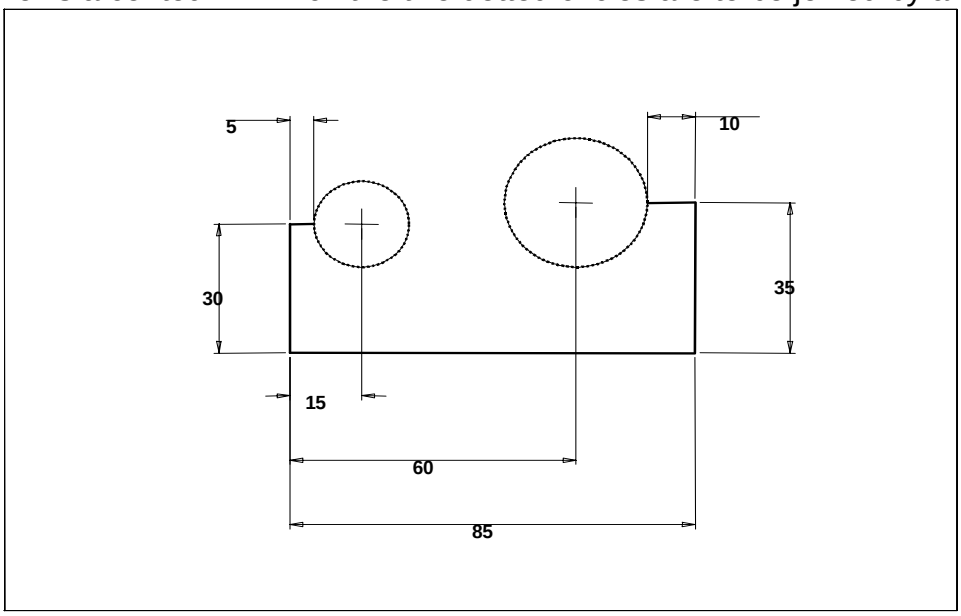


Figure 21

Figure 22 shows the four different shapes that can be produced, simply by changing the directions of the two arcs.

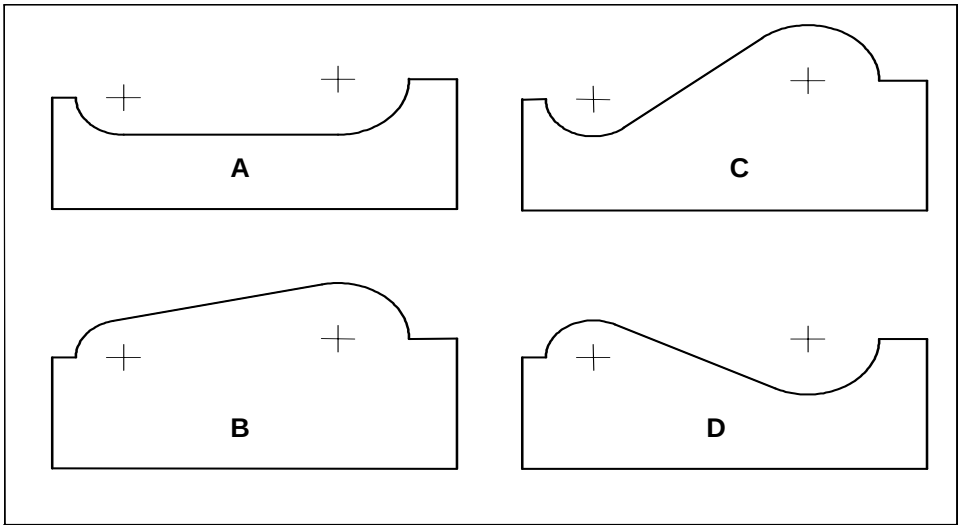


Figure 22

The programs for each of the shape is show below. All the shapes have been programmed in a generally CCW direction starting from the lower left corner.

In Figure 22A both arcs are clockwise

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	85.000	0.000	0.000	0.000	0.000	0.000
3	L	0	85.000	35.000	0.000	0.000	0.000	0.000
4	L	0	75.000	35.000	0.000	0.000	0.000	0.000
5	CW	1	75.000	35.000	0.000	-15.000	0.000	0.000
6	L	8	0.000	0.000	0.000	0.000	0.000	0.000
7	CW	10	5.000	30.000	0.000	10.000	0.000	0.000
8	L	0	0.000	30.000	0.000	0.000	0.000	0.000
E 9	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Figure 22A

The first 4 Steps are simple geometry requiring no special functions.

Step 5 the end point of the arc is the same as its start point. This end-point is temporary and will be replaced by the point at which the arc is tangent to the line in step 6. ***Note*** because step 6 define a line tangent to ***2*** arcs, its X & Y parameters are ignored.

Step 8 defines the second arc. Its start point is not known, so FN10 is used to specify that the I and J parameters are measured from the arc end-point.

The remainder of the contour can be defined using simple Line functions.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	85.000	0.000	0.000	0.000	0.000	0.000
3	L	0	85.000	35.000	0.000	0.000	0.000	0.000
4	L	0	75.000	35.000	0.000	0.000	0.000	0.000
5	CCW	1	75.000	35.000	0.000	-15.000	0.000	0.000
6	L	8	0.000	0.000	0.000	0.000	0.000	0.000
7	CCW	10	5.000	30.000	0.000	10.000	0.000	0.000
8	L	0	0.000	30.000	0.000	0.000	0.000	0.000
E 9	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Figure 22B

Reversing the direction of the arcs produces Figure 22B.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	85.000	0.000	0.000	0.000	0.000	0.000
3	L	0	85.000	35.000	0.000	0.000	0.000	0.000
4	L	0	75.000	35.000	0.000	0.000	0.000	0.000
5	CCW	1	75.000	35.000	0.000	-15.000	0.000	0.000
6	L	8	0.000	0.000	0.000	0.000	0.000	0.000
7	CW	10	5.000	30.000	0.000	10.000	0.000	0.000
8	L	0	0.000	30.000	0.000	0.000	0.000	0.000
E 9	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Figure 22C

Reversing just one of them produces 22C

And reversing the other arc produces Figure 22D

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	85.000	0.000	0.000	0.000	0.000	0.000
3	L	0	85.000	35.000	0.000	0.000	0.000	0.000
4	L	0	75.000	35.000	0.000	0.000	0.000	0.000
5	CW	1	75.000	35.000	0.000	-15.000	0.000	0.000
6	L	8	0.000	0.000	0.000	0.000	0.000	0.000
7	CCW	10	5.000	30.000	0.000	10.000	0.000	0.000
8	L	0	0.000	30.000	0.000	0.000	0.000	0.000
E 9	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Figure 22D

Tangents and Fillets

In the same way that Fillets or Chamfers can be included between intersecting Lines and Arcs, so they can between tangent lines and arcs. But there is a catch …

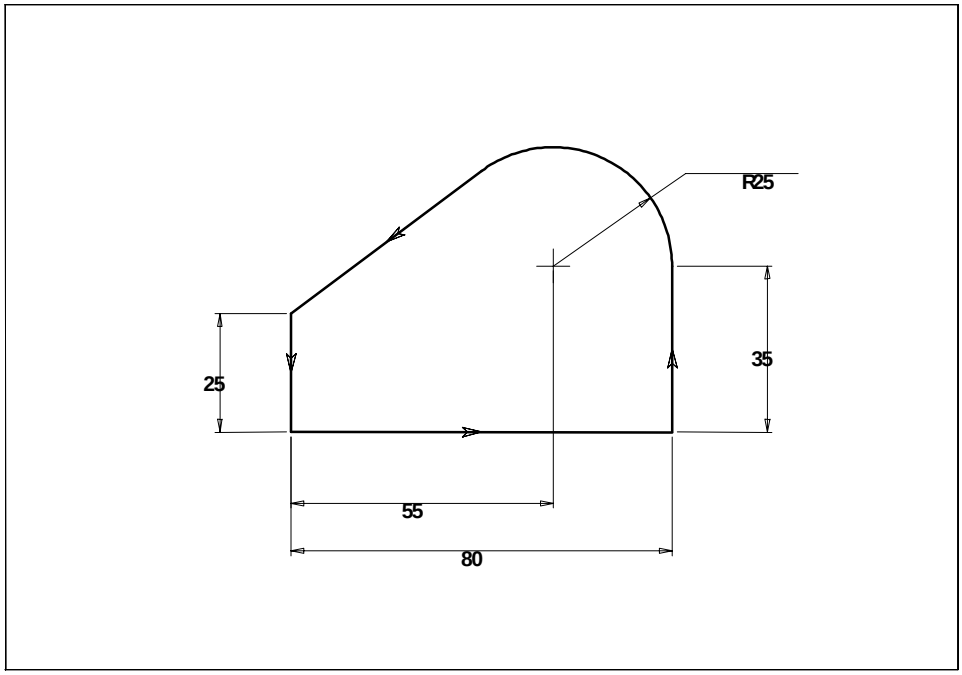


Figure 23

Looking at Figure 23 – which is the same as Figure 20 – it is not possible to insert a Fillet between R25 arc and the angled line because they are tangential. If a fillet is programmed beween the arc and the line, then the line must be repositioned to accomodate the fillet, this is achieved by using the alternate tangent line as show below.

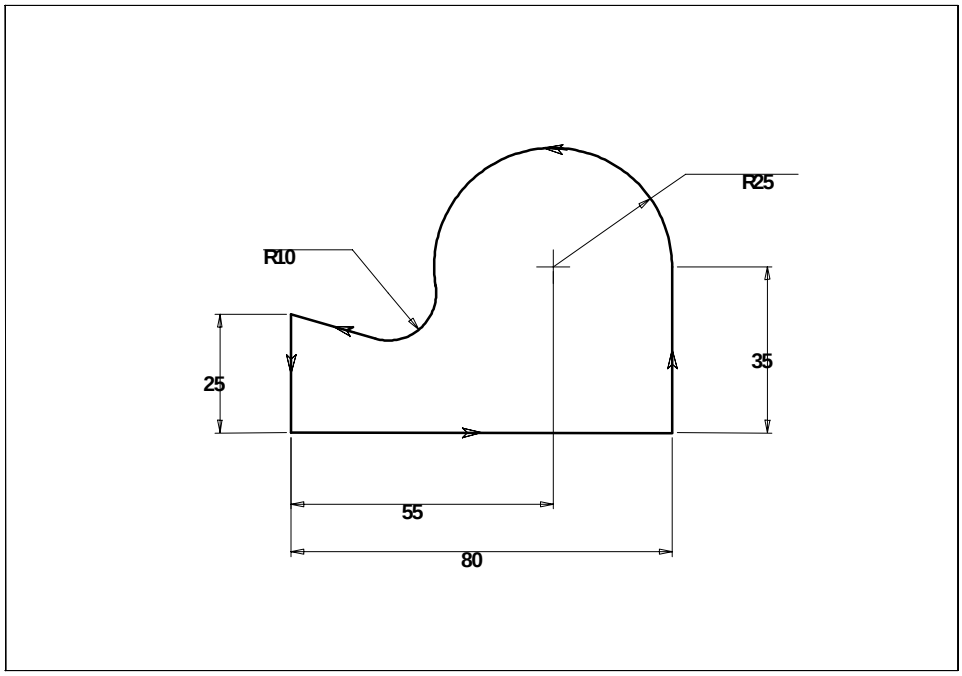


Figure 24

To program this shape, use the program for Figure 20, and add enter a value of 10 in the Radius column on step 4.

Defining Arcs tangent to a fixed line or arc

The arc tangent functions 4 thro' 9, provide a means of programming an arc tangent to a fixed object. In each case the centre point of the arc will be unknown as will one or both end points. I and J parameters are ignored and the arc radius must be entered in the Radius column.

Functions 4, 5, 6 and 7 define arcs in which one end is fixed and the other is defined as the tangent point with the previous or following Line or Arc step. If the ***fixed*** object that the arc is tangent to, is a line, it must be programmed using FN4 or FN5.

Arc Functions 4 & 5 *Specifies an arc which* ***starts*** *at a known point and is tangent to the Line or Arc step which follows. The XY columns contain the start point of the arc and the Radius column contains its radius. The I and J values are ignored. If two solutions are possible, Function 4 selects the minor arc and Function 5 selects the major arc.*

If the following step is an arc, then a negative value in the Radius column will cause one arc to be ***inside*** *the other, while a positive value will ensure that the two arcs are both* ***outside*** *each other.*

Arc Functions 6 & 7 *Specifies an arc which is tangent to the Line or Arc step which precedes it, and* ***ends*** *at a known point and. The XY columns contain the end point of the arc and the Radius column contains its radius. The I&J values are ignored. If two solutions are possible, Function 6 selects the minor arc and Function 7 selects the major arc.*

If the preceding step is an arc, then a negative value in the Radius column will cause one arc to be ***inside*** *the other, while a positive value will ensure that the two arcs are both* ***outside*** *each other.*

Arc Functions 8 & 9 Specifies an arc tangent to both the previous and the next elements. Only the Radius column has data. X, Y, I & J are all ignored.

Tangent arc examples – Arc tangent to a fixed line

Figure 25 shows a contour which is similar to figure 20. The main difference is that in Figure 20 the position of the arc is fixed but in figure 25 it is the angled line that is fixed, and the arc's end and centre point are unknown.

As an added complication there are two arcs of R25mm which are tangential to the angles line and which pass through the point X80 Y35 (i.e the arc start point). The desired arc is indicated by the Function number attached to the arc step as shown below.

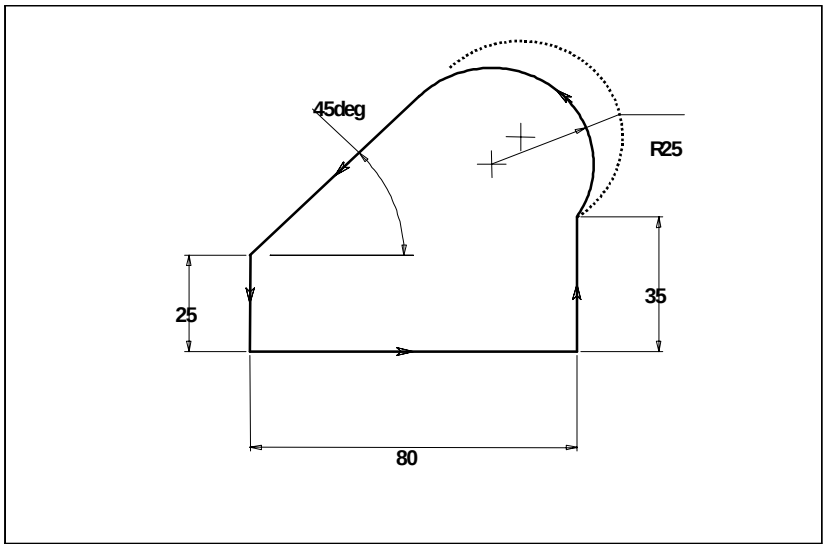


Figure 25

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	80.000	0.000	0.000	0.000	0.000	0.000
3	L	0	80.000	35.000	0.000	0.000	0.000	0.000
4	CCW	4	80.000	35.000	0.000	0.000	0.000	25.000
5	L	4	0.000	25.000	0.000	-135.000	0.000	0.000
6	L	0	0.000	0.000	0.000	0.000	0.000	0.000

The program for figure 25

Steps 1, 2 and 3 are simple geometry requiring no special functions.

Step 4 uses FN4 to define a CCW arc which starts at the point X80Y35, and has Radius 25mm. The arc ends at the tangent point with the line defined in step 5. (To select the dotted arc use FN5 instead of FN4) No I and J values can be programmed for this arc because its centre point is not known. The arc's radius must be entered into the Radius column.

Step 5 defines a line at an angle of –135 degrees to the X axis which passes through the point X0 Y25.

Step 6 is a simple line back to the start point.

Programming Figure 25 in the reverse direction produces the following program

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	0.000	25.000	0.000	0.000	0.000	0.000
3	L	4	0.000	25.000	0.000	45.000	0.000	0.000
4	CW	6	80.000	35.000	0.000	0.000	0.000	25.000
5	L	0	80.000	0.000	0.000	0.000	0.000	0.000
6	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Figure 25, Reversed

The main difference when the shape is reversed is that the arc is now CW instead of CCW, and uses FN6 instead of FN4. (FN6 defines an arc tangent to the preceding Line or Arc step, which ends at a known point.)

Tangent arc examples – Arc tangent to a fixed arc

Programming a ***floating*** arc through a point, and tangent to a ***fixed*** arc present some new problems.

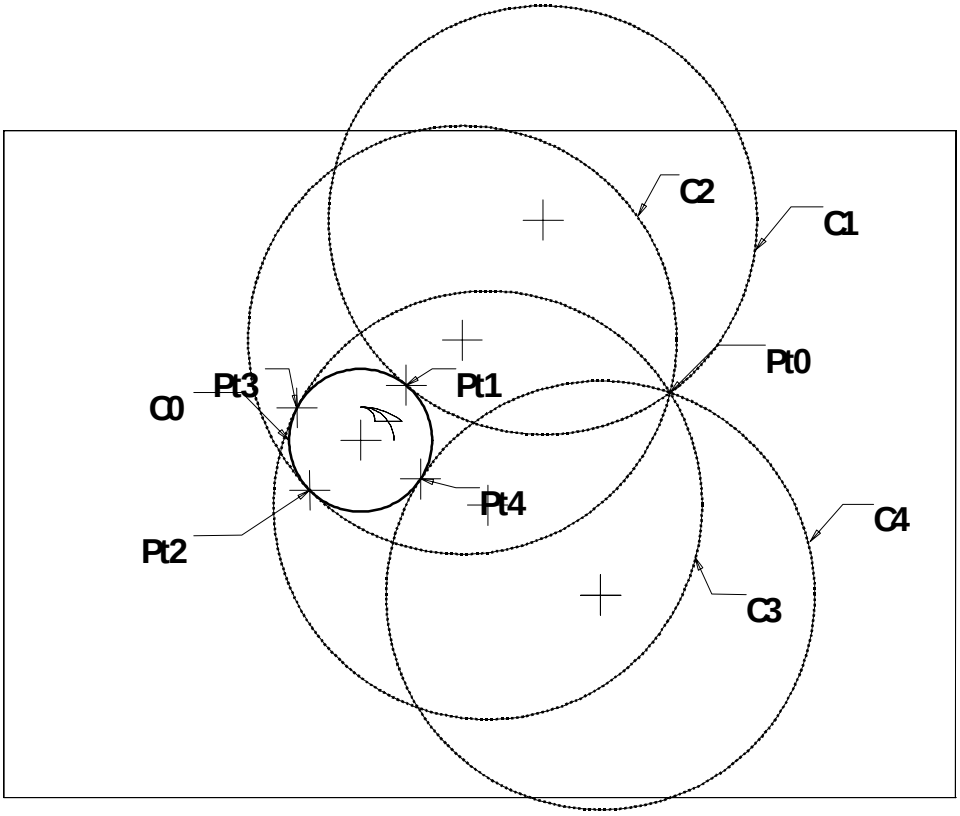


Figure 26

Figure 26 shows 4 circles which pass through the point Pt0 and are all tangent to the circle C0.

The circle C1 is tangent to circle C0 at the point Pt1, C2 is tangent at Pt2 and so-on.

Considering the circle C1, this can be split into two arcs by splitting the circle at the point Pt0 and again at Pt1. Since each of the 4 circles can be split into two arcs, there is a total of 8 solutions to the task of finding an arc through a point, tangent to a fixed arc.

Remember that there are two pairs of tangent functions ..

FN4 and FN5 find an arc starting at a fixed point tangent to the arc or line in the following step. FN4 and FN5 differentiate the major and minor arcs. *(When a circle is split into two arcs, one will be longer than the other, unless they are both semi-circles, this is the major arc, selected using FN5. The other shorter arc is the minor arc selected using FN4)*

(FN6 and FN7 are used when the end point of the arc is known and its start point must be calculated.)

The second mechanism used to differentiate the arcs is the sign of the arc radius. If the radius is entered as a positive value, then the selected *floating* arc and the *fixed arc* will be “outside” each other, as in C1 and C4. If the arc radius is programmed as a negative value then one arc will be “inside” the other as in C2 and C3.

Finally the arc directions are also important. Comparing circles C1 and C2 it can be seen that they must have opposite directions.

Putting all this together to select the shortest arc on circle C1, the arc must be programmed using FN4 - to select the minor arc, it must be CCW because arc C0 is CW and the radius must be positive because C0 is “outside” C1.

A practical example …

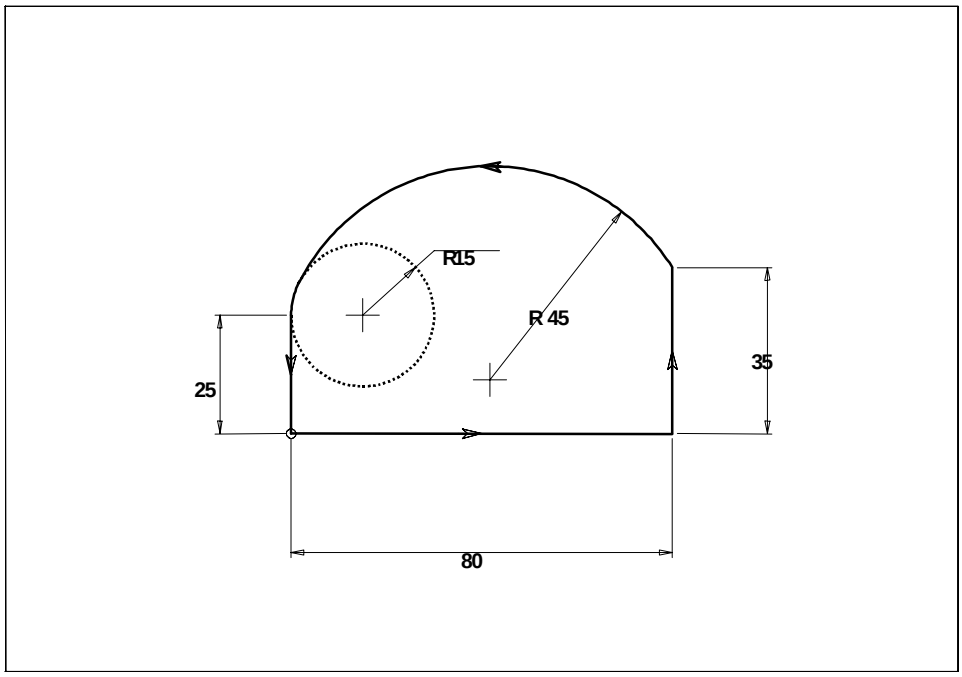


Figure 27

In Figure 27 all that is know about the R45 arc is its start point, radius and direction. The R15 arc has a known centre point, end point and direction. The program is simple …

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	80.000	0.000	0.000	0.000	0.000	0.000
3	L	0	80.000	35.000	0.000	0.000	0.000	0.000
4	CCW	4	80.000	35.000	0.000	0.000	0.000	-45.000
5	CCW	10	0.000	25.000	0.000	15.000	0.000	0.000
E 6	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Steps 1, 2, and 3 are simple geometry with no special functions.

Step 4 uses FN4 because the R45 arc is the minor arc, and has a negative radius the arc must be “outside” the R15 arc. The arc has been programmed with X80 Y35 as the end – point, but could be programmed with any value as the X, Y I and J parameters are all ignored.

Step 5 uses the familiar FN10 to indicate that the start point is unknown and that the I and J parameters are measured from the arc endpoint.

Arc tangent to two fixed arcs

The arc tangent to two fixed arcs is similar to the line tangent to two arcs problem. The system does not support the concept of “inside” and “outside” arcs (as with function 4, 5, 6 & 7). It simply uses the direction of the three arcs to determine the most reasonable solution. I.e. the one that maintains continuity of direction at the tangent points. This only leaves the programmer to decide whether to use FN8 for the minor arc or FN9 for the major arc.

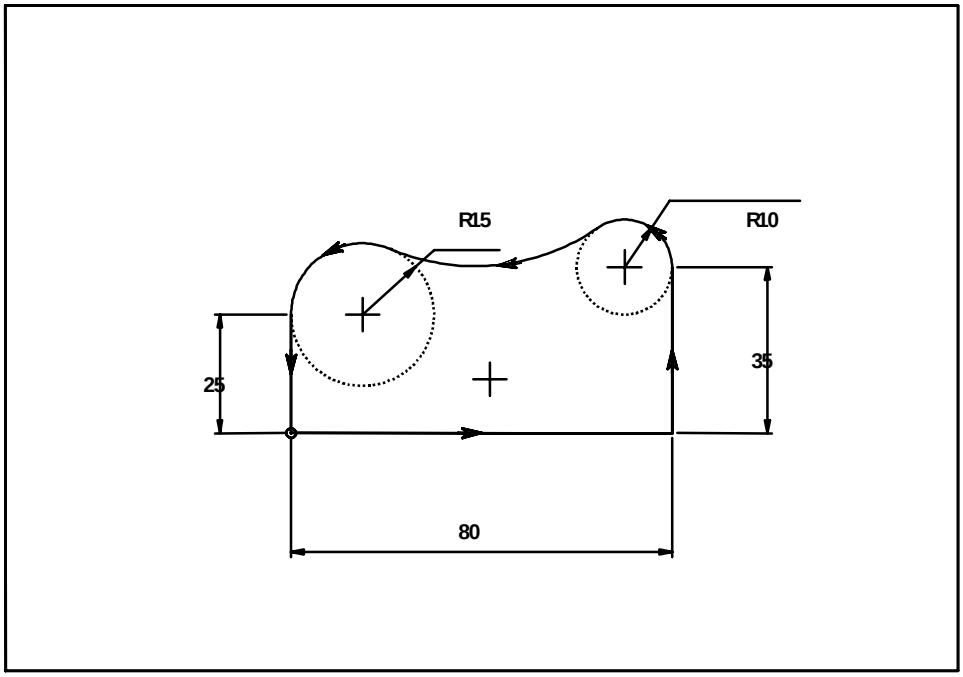


Figure 28

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	80.000	0.000	0.000	0.000	0.000	0.000
3	L	0	80.000	35.000	0.000	0.000	0.000	0.000
4	CCW	1	80.000	35.000	0.000	-10.000	0.000	0.000
5	CW	8	0.000	0.000	0.000	0.000	0.000	45.000
6	CCW	10	0.000	25.000	0.000	15.000	0.000	0.000
E 7	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Program for Figure 28

Figure 10 holds no surprises. Steps 1, 2, and 3 are all simple geometry requiring no special functions.

Step 4 is a simple arc using Fn1. The end-point is X80 Y35, creating a complete circle. This end point is temporary and will be replaced by the tangent point with the arc defined in step 5.

Step 5 defines a clockwise arc using function 8. The X, Y I and J parameters are all ignored, only the Radius is significant. The arcs start and end points will be the points of tangency with the arcs in steps 4 and 6.

Step 6 is an arc defined using FN10 (because the start point is not known). I and J are relative to the end point. Step 6 is a simple line back to the start of the contour.

Arc tangent to a fixed line, passing through 2 fixed points

There is one final pair of tangent arc functions, FN11 and FN12.

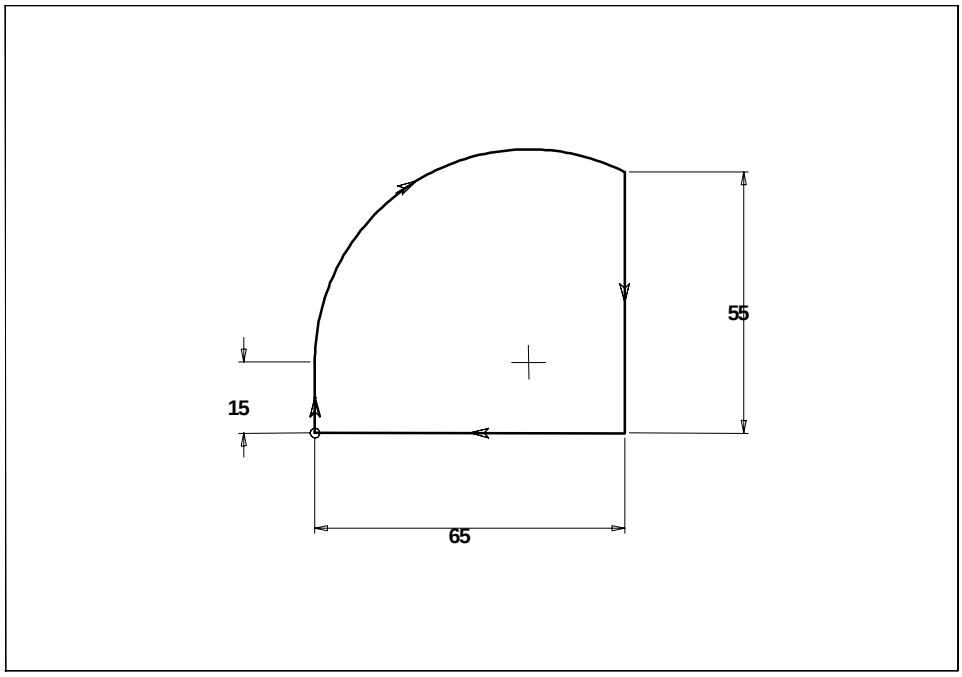


Figure 29

Figure 29 shows a deceptively simple contour. The CW arc has very little data, its radius and centre point are not known. All that is known about this arc is that it starts at X0Y15 and ends at X65 Y55, it has CW direction, and it is tangential to the left hand vertical line.

Arc FN11 *defines an arc which is tangential to the preceding Line at a known point, and which ends at a known point.*

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	0.000	0.000	0.000	0.000	0.000	0.000
2	L	0	0.000	15.000	0.000	0.000	0.000	0.000
3	CW	11	65.000	55.000	0.000	0.000	0.000	0.000
4	L	0	65.000	0.000	0.000	0.000	0.000	0.000
E 7	L	0	0.000	0.000	0.000	0.000	0.000	0.000

Program for Figure 29

The only part of program 29 that needs explanation is the CW arc in step 3.

This arc uses FN11 to define an arc tangential to the preceding line, at the ***end point*** of that line, and ending at a known point – X65 Y55. The I J and Radius parameters are all ignored.

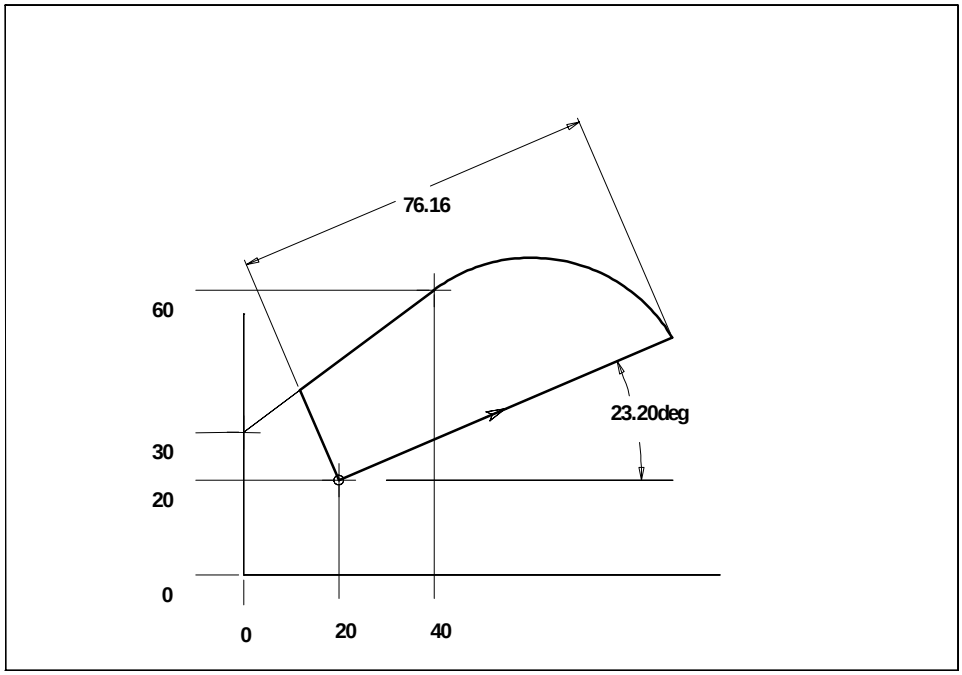


Figure 29a

The program for Figure 29a will use Arc Fn12 to define the arc segment, and will introduce the use of Polar coordinate systems. *Polar coordinate systems are described fully in the next section – The Point functions.*

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	20.000	20.000	0.000	0.000	0.000	0.000
2	P	3	0.000	0.000	0.000	0.000	0.000	0.000
3	L	0	76.160	23.20	0.000	0.000	0.000	0.000
4	P	4	0.000	0.000	0.000	0.000	0.000	0.000
5	CCW	12	40.000	60.000	0.000	0.000	0.000	0.000
6	L	2	0.000	30.000	0.000	0.000	0.000	0.000
7	L	4	20.000	20.000	0.000	-66.800	0.000	0.000

Program for Figure 29a

Step 1 defines the start of the Contour at the point X20 Y20.

Step 2 uses Point Fn3 to switch on Polar coordinate system. The X and Y values are the position of the new pole, incremental from the current position. X0 Y0 makes the current position the pole.

Step 3 defines a line in polar coordinates. The radius is entered into the X column and the angle is entered into the Y column.

Step 4 uses Point Fn4 to switch off polar coordinate programming.

Step 5 uses Fn12 to define an arc which starts at the end point defined in step 3, and ends at the X Y coordinate X40 Y60. The position of the arc centre and its radius are not known, but the arc will be tangential to the line defined in step 6.

Step 6 uses Fn2 to define a line through 2 points. I.e. the end point defined in step 5, and the point X0 Y30. This end point is not on the contour, and will be replaced when the intersection point with this line, and the line defined in step 7, is calculated by the system.

Finally Step 7 uses Fn4 to define a line at an angle of –66.80 degrees which ends back at the start point X20 Y20.

Coordinate systems

SO far all the discussion and examples have been based on a Cartesian (X, Y) coordinate system.

The point functions can be used to define alternate coordinate systems.

The Point functions

Function 0

Function 0 defines a "*Move point*" in the current coordinate system. Each contour programmed must start with a *Move point*.

Function 1

Function 1 defines an *auxiliary point*. An auxiliary point is one which is used purely to simplify the programming of the next step. It defines a temporary start point for the next line or arc. The auxiliary point need not be – and typically wont be – on the actual contour.

Function 2

Function 2 enables Polar coordinate programming, and defines the absolute position of the new coordinate system origin, in the current coordinate system. Using FN2 the pole is defined absolute from X0 Y0.

Function 3

Function 3 enables Polar coordinate programming, and defines the incremental position of the new coordinate system origin, in the current coordinate system. Using FN3 the pole is defined relative to the current position.

Function 4

Switches back to Cartesian coordinates, and resets the coordinate system origin to X0 Y0. Function 4 cancels any new coordinate system set up using FN3 or FN5.

Auxiliary points

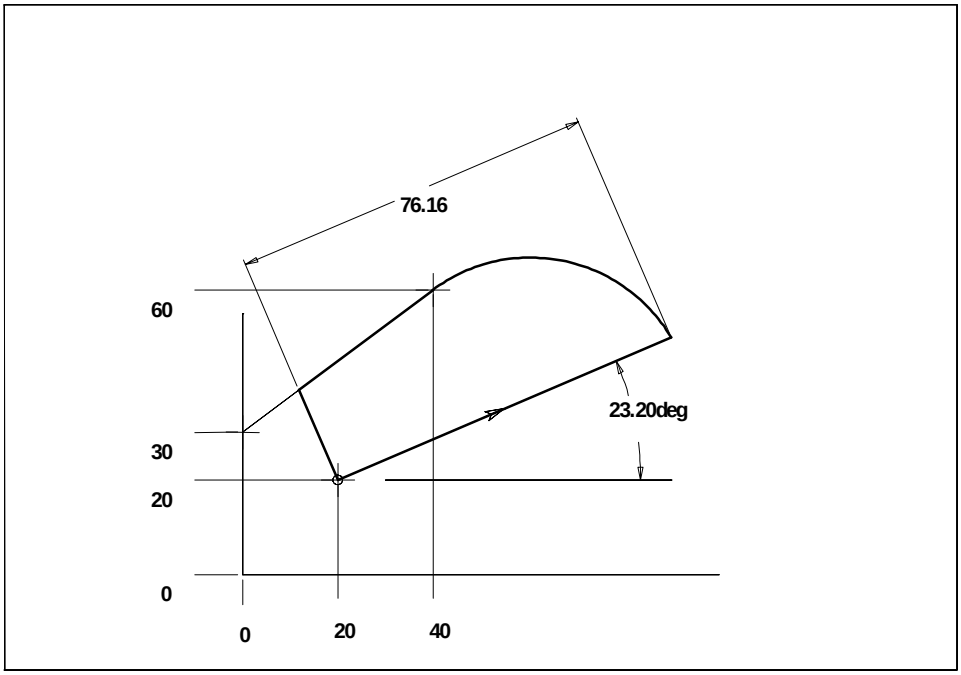


Figure 29b

The program for Figure 29b is actually the same as 29a, but instead of programming it in the direction shown, the program shown below defines the shape in a generally CW direction.

Step	Cmd	Fn	X	Y	Z	I	J	R
1	P	0	20.000	20.000	0.000	0.000	0.000	0.000
2	L	4	20.000	20.000	0.000	113.200	0.000	0.000
3	P	1	0.000	30.000	0.000	0.000	0.000	0.000
4	L	0	40.000	60.000	0.000	0.000	0.000	0.000
5	P	2	20.000	20.000	0.000	0.000	0.000	0.000
6	CW	11	76.160	23.200	0.000	0.000	0.000	0.000
7	P	4	0.000	0.000	0.000	0.000	0.000	0.000
E 8	L	0	20.000	20.000	0.000	0.000	0.000	0.000

Program for Figure 29b

1. Define the start point at X20, Y20
2. Define a line at an angle of 113.20, passing through the start point.
3. Define an ***auxiliary point*** at X0 Y30, which acts as a temporary start point for the line in the next step.
4. Define a line to the point X40, Y60. The start point of this line will be its intersection with the line is step2.
5. Define a polar coordinate system origin at X20 Y20.
6. Define an arc tangent to the line in step 4, which starts at X40, Y60 and ends at the polar coordinates PR76.16, PA 23.20, relative to the pole at X20, Y20.
7. Switch off polar coordinates
8. Define a line back to the start point X20, Y20.

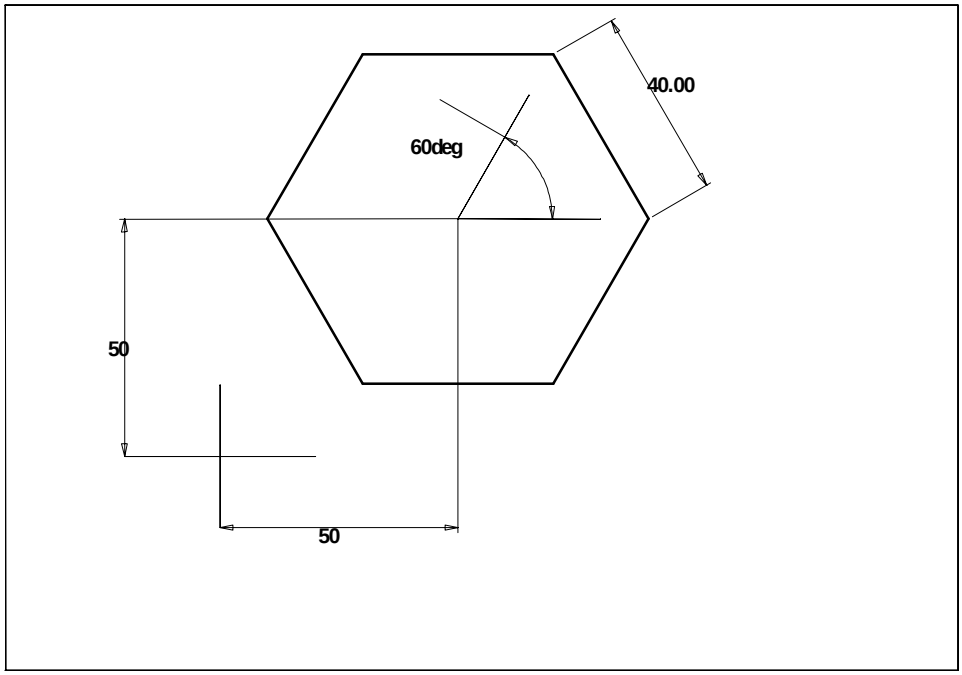


Figure 30

Figure 30 shows a hexagonal contour most conveniently programmed using polar coordinates. Each vertex is at a distance of 40mm from the centre. Using Point Fn2 a new program origin can be defined at the centre of the shape …

When using polar coordinates, the Polar Radius value is entered in the X column, and the Polar Angle is entered into the Y column.

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	2	50.000	50.000	0.000	0.000	0.000	0.000
2	P	0	40.000	0.000	0.000	0.000	0.000	0.000
3	L	0	40.000	60.000	0.000	0.000	0.000	0.000
4	L	0	40.000	120.000	0.000	0.000	0.000	0.000
5	L	0	40.000	180.000	0.000	0.000	0.000	0.000
6	L	0	40.000	240.000	0.000	0.000	0.000	0.000
7	L	0	40.000	300.000	0.000	0.000	0.000	0.000
E 8	L	0	40.000	360.000	0.000	0.000	0.000	0.000

Using Polar Coordinates to program Figure 30

Step 1 uses FN2 to define a polar coordinate system origin at the point X50 Y50.

Step 2 defines a move point, or the start of the contour, it is at a radius of 40, Angle 0 from the polar origin.

Steps 3 thro' 8 define each line making up the hexagon. For each end point the radius is a constant 40mm, only the angle changes, increasing by 60 degrees at each vertex.

Summary

The start and end point of each line or arc segment can be defined in several ways …

Defining the start point of a line or arc

There are three ways to define the start point of a line or arc …

1. By default system assumes that the start point of a line or arc segment coincides with the end point of the previous segment.
2. The start point can be defined to be the intersection point of the previous line or arc segment, and the current line or arc segment. This also implicitly defines the end point of the previous segment.
3. The start point can be defined to be the point of tangency between the previous line or arc segment and the current line or arc segment. As with points found by intersection, if the start point of one segment is defined as it tangent point with the previous segment, then the end point of the previous segment is also defined.

Defining the end point of a line or arc.

There are also three ways to define the end point of a line or arc …

1. By default the end point is defined explicitly using either…
    a. XY Cartesian coordinate position absolute from X0, Y0, or incremental from the previous XY coordinate. Or …
    b. RA Polar coordinate position absolute from the current polar origin.
2. The end point of a line or arc can be defined to be the point at which it intersects the next line or arc segment.
3. The end point of a line or arc can defined as the point of tangency with the next line or arc segment.

An arc segment requires two additional pieces of information, the centre point and hence the radius, and the direction either CW or CCW.

Defining arc centre points

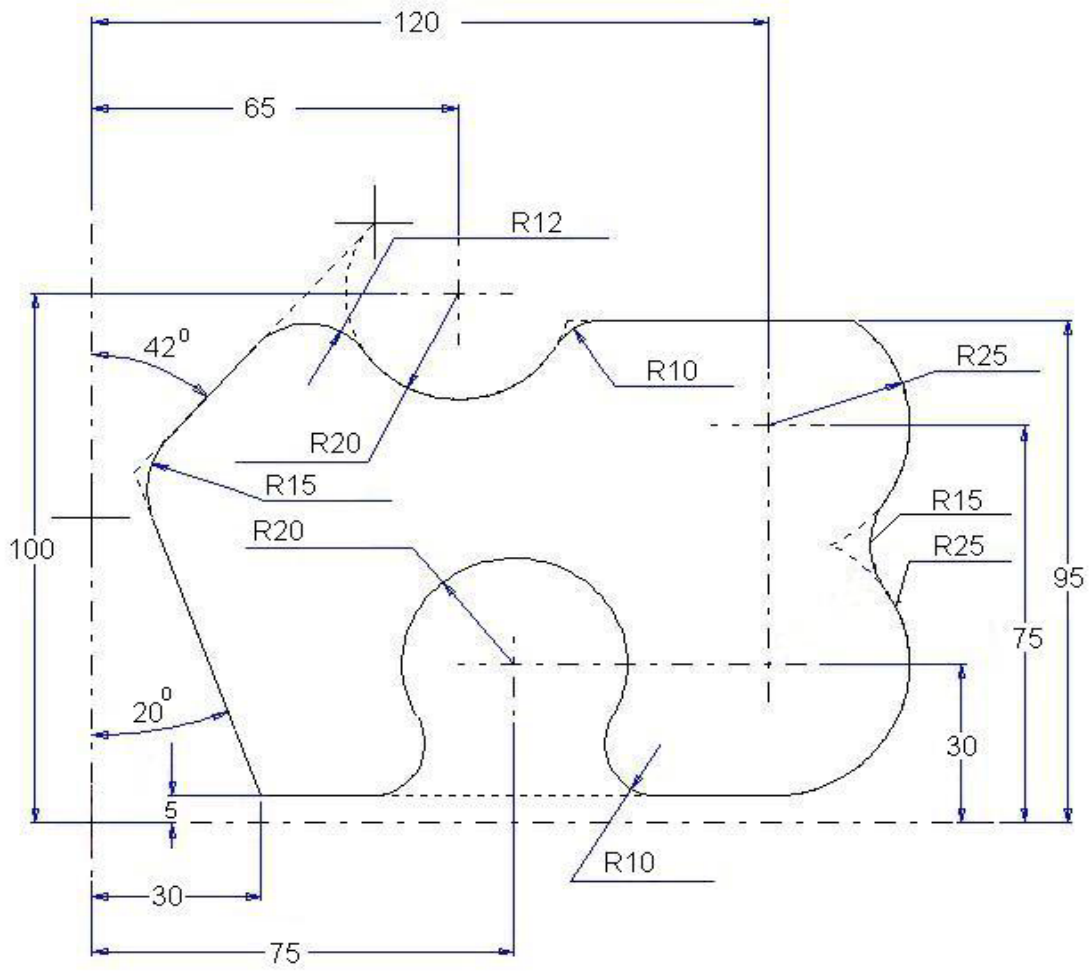
Arc centre points can be define in xxx ways …

1. If the start point of the arc is known, the arc centre can be defined explicitly using I and J values to represent the distance in the X and Y axes from the start point of the arc to the centre of the arc.
2. If both the start and end points are known, then instead of defining the centre using I and J values, the radius can be defined, in which case system will calculate the centre point.
3. If the start point is unknown, then the centre can be defined using I and J values measured from ***any*** convenient point on the arc. This convention can only be used when the start point (of the arc) is to be calculated as an intersection point or tangent point.

If an arc is being defined as tangent to either the previous or the following line or arc, then system must calculate the arc centre point, and any values entered in the I and J columns are ignored.

More example

Program : 801

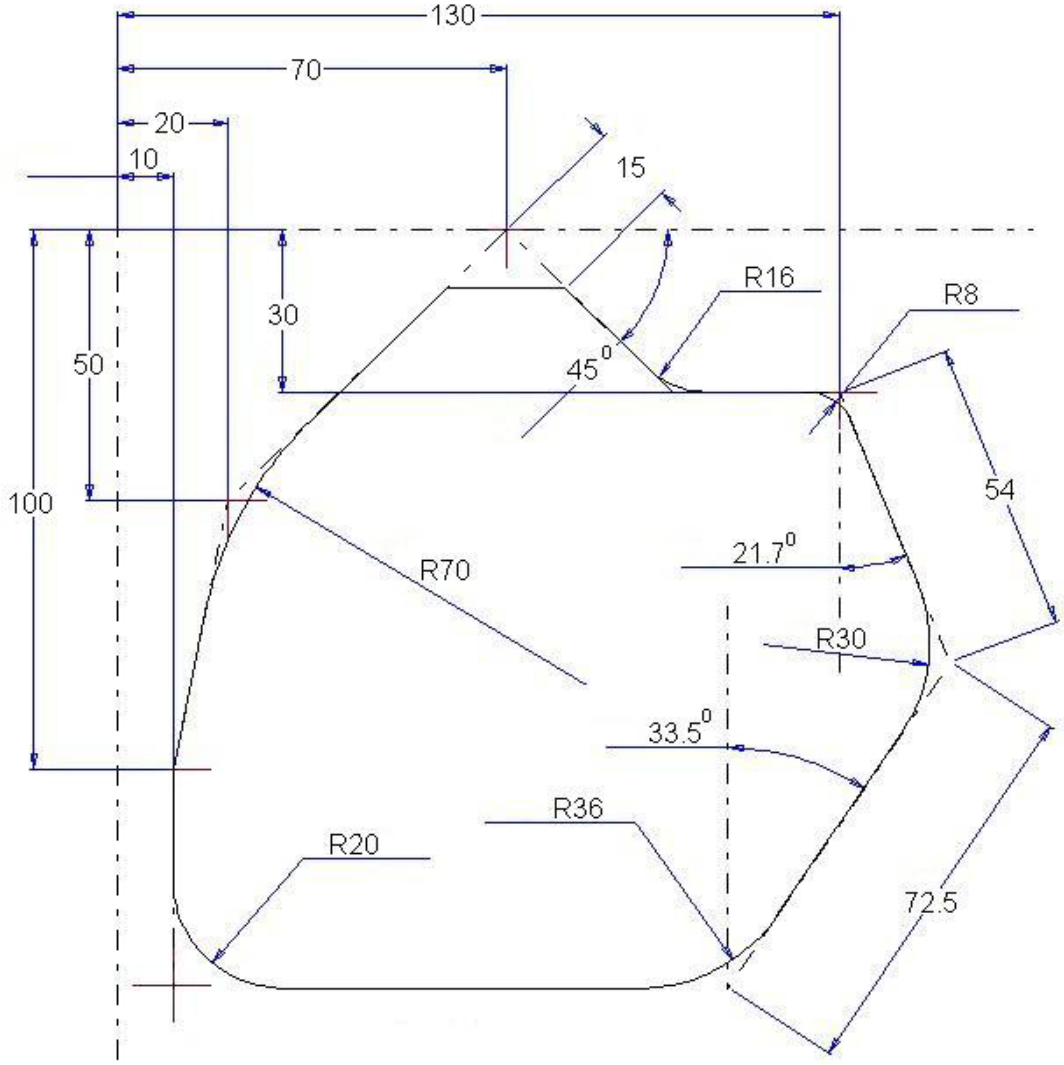


EDIT PROGRAM NO.: 801 (820) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	N/R
1	P	0	+40.000	+5.000	+0.000	+0.000	+0.000	+0.000
2	L	0	+30.000	+5.000	+0.000	+0.000	+0.000	+0.000
3	L	4	+30.000	+5.000	+0.000	+110.000	+0.000	+15.000
4	L	9	+0.000	+0.000	+0.000	+42.000	+0.000	+12.000
5	CCW	10	+65.000	+80.000	+0.000	+0.000	+20.000	+0.000
6	CCW	2	+65.000	+80.000	+0.000	+0.000	+20.000	+10.000
7	L	4	+120.000	+95.000	+0.000	+0.000	+0.000	+0.000
8	L	4	+120.000	+95.000	+0.000	+0.000	+0.000	+0.000
9	CW	2	+145.000	+75.000	+0.000	-25.000	+0.000	+25.000
10	CW	10	+120.000	+5.000	+0.000	+0.000	+25.000	+0.000
11	L	1	+40.000	+5.000	+0.000	+180.000	+0.000	+0.000
12	CW	9	+0.000	+0.000	+0.000	+0.000	+0.000	+10.000
13	CCW	10	+75.000	+50.000	+0.000	+0.000	-20.000	+0.000
14	CW	8	+0.000	+0.000	+0.000	+0.000	+0.000	+10.000
15	P	1	+50.000	+5.000	+0.000	+0.000	+0.000	+0.000
E 16	L	0	+40.000	+5.000	+0.000	+0.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 802

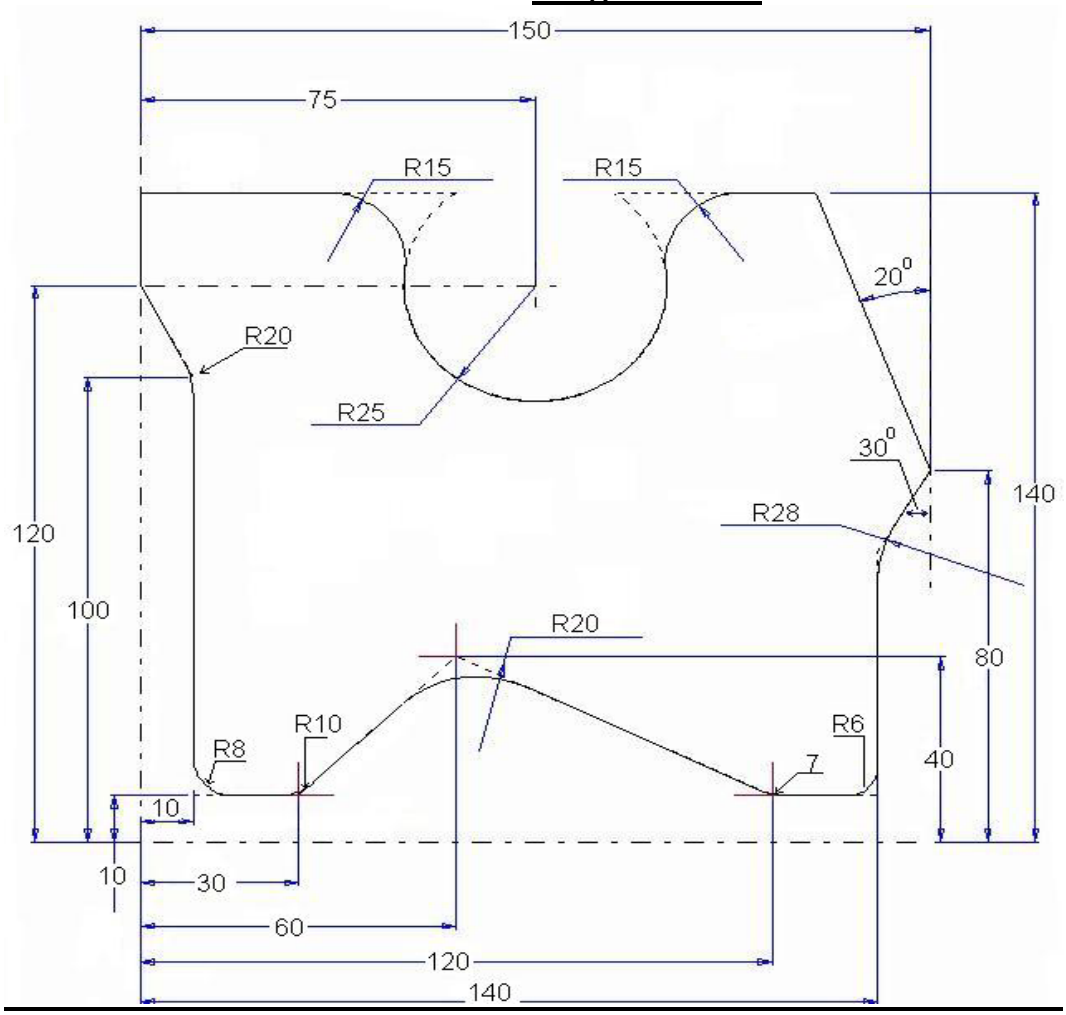


EDIT PROGRAM NO.: 802 (802) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	N/R
1	P	0	+10.000	-100.000	+0.000	+0.000	+0.000	+0.000
2	L	0	+20.000	-50.000	+0.000	+0.000	+0.000	+70.000
3	L	0	+70.000	+0.000	+0.000	+0.000	+0.000	-15.000
4	L	4	+70.000	+0.000	+0.000	-45.000	+0.000	+16.000
5	L	4	+130.000	-30.000	+0.000	+0.000	+0.000	+8.000
6	P	3	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
7	L	0	+54.000	-68.300	+0.000	+0.000	+0.000	+30.000
8	P	3	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
9	L	0	+72.500	+236.500	+0.000	+0.000	+0.000	+30.000
10	L	4	+72.500	+236.500	+0.000	+180.000	+0.000	+20.000
11	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
E 12	L	4	+10.000	-100.000	+0.000	+90.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 803

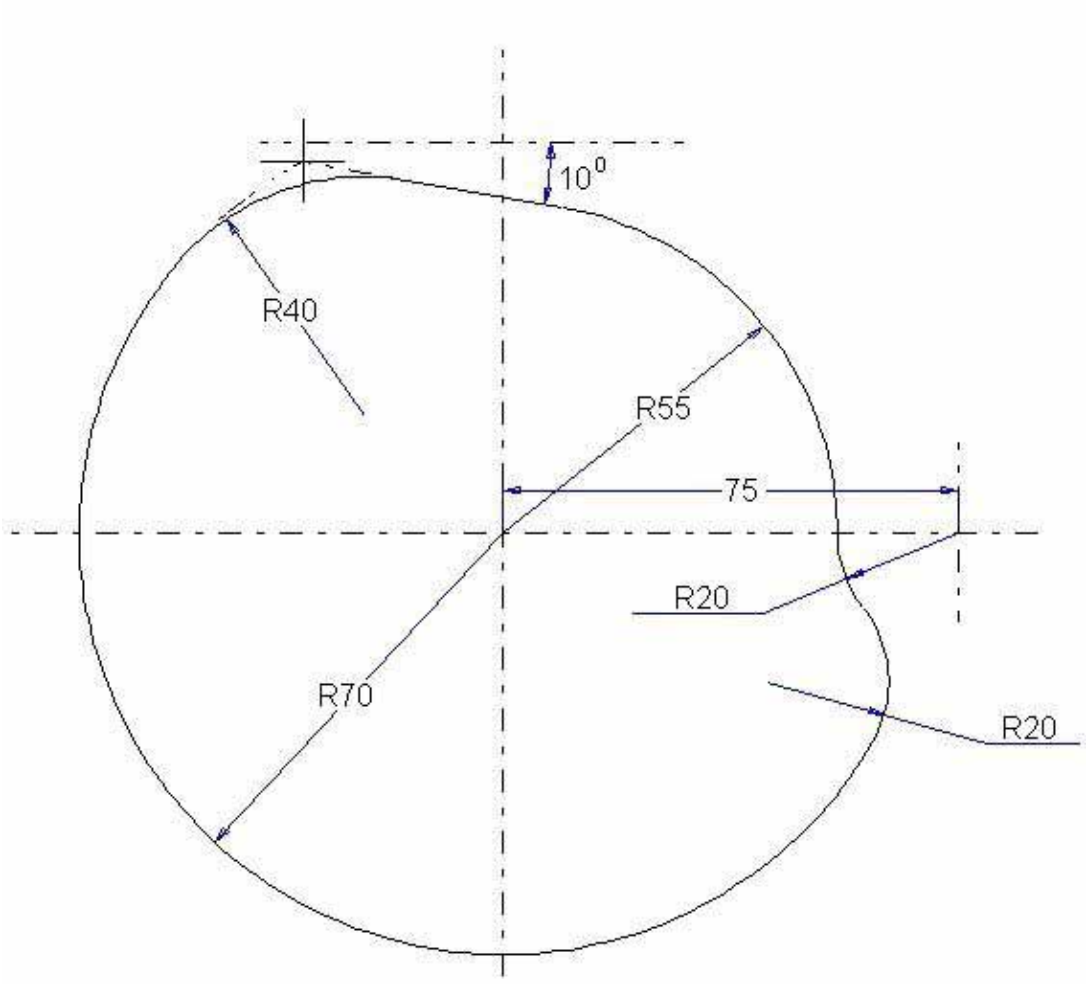


EDIT PROGRAM NO.: 803 (802) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	R+/C-
1	P	0	+0.000	+140.000	+0.000	+0.000	+0.000	+0.000
2	L	3	+75.000	+140.000	+0.000	+0.000	+0.000	+15.000
3	CCW	10	+75.000	+95.000	+0.000	+0.000	+25.000	+0.000
4	CCW	2	+75.000	+95.000	+0.000	+0.000	+25.000	+15.000
5	L	4	+120.000	+140.000	+0.000	+0.000	+0.000	+0.000
6	L	4	+120.000	+140.000	+0.000	+0.000	+0.000	+0.000
7	L	4	+150.000	+80.000	+0.000	-70.000	+0.000	+0.000
8	L	4	+150.000	+80.000	+0.000	+240.000	+0.000	+28.000
9	L	4	+140.000	+10.000	+0.000	+270.000	+0.000	+6.000
10	L	0	+120.000	+10.000	+0.000	+0.000	+0.000	-7.000
11	L	0	+60.000	+40.000	+0.000	+0.000	+0.000	+20.000
12	L	0	+30.000	+10.000	+0.000	+0.000	+0.000	+10.000
13	L	0	+10.000	+10.000	+0.000	+0.000	+0.000	+8.000
14	L	0	+10.000	+100.000	+0.000	+0.000	+0.000	+20.000
15	L	0	+0.000	+120.000	+0.000	+0.000	+0.000	+0.000
E 16	L	0	+0.000	+140.000	+0.000	+0.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 804

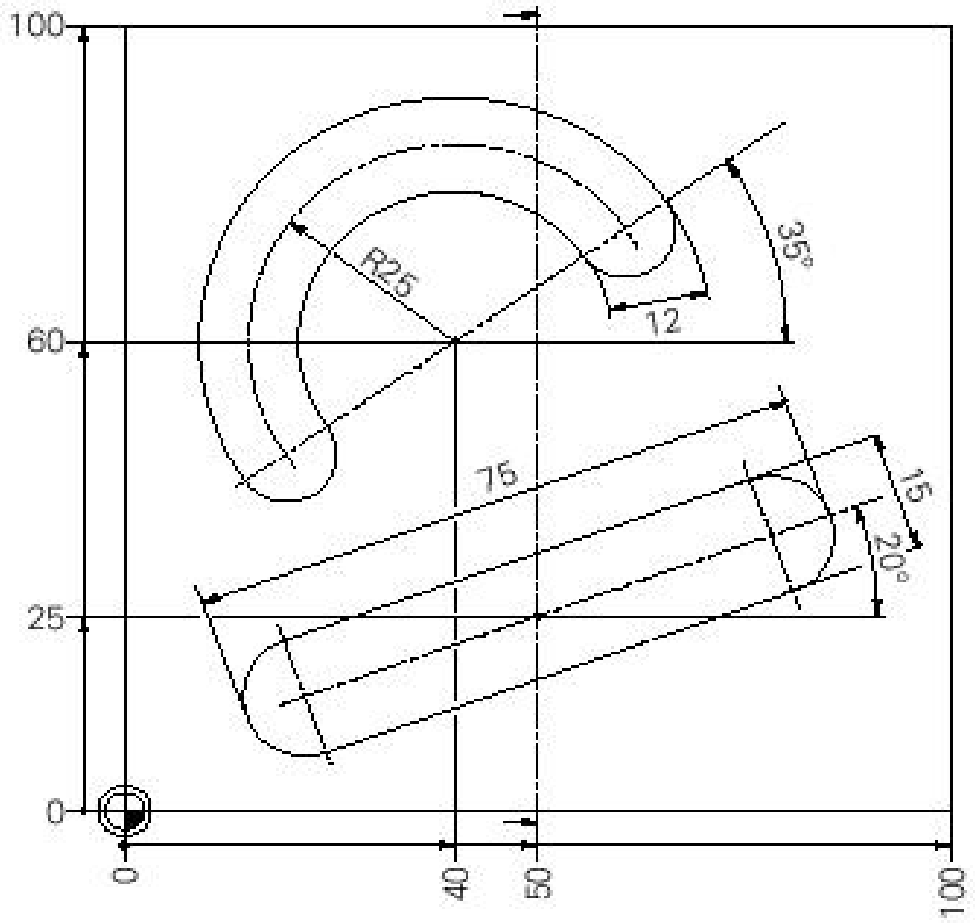


EDIT PROGRAM NO.: 804 (803) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	N/R
1	P	2	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
2	P	0	+70.000	-90.000	+0.000	+0.000	+0.000	+0.000
3	CW	3	+70.000	+90.000	+0.000	+0.000	+0.000	+40.000
4	L	9	+0.000	+0.000	+0.000	-10.000	+0.000	+0.000
5	CW	10	+55.000	+0.000	+0.000	+0.000	+0.000	+0.000
6	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
7	CCW	2	+55.000	+0.000	+0.000	+20.000	+0.000	+20.000
E 8	CW	10	+0.000	-70.000	+0.000	+0.000	+70.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 811

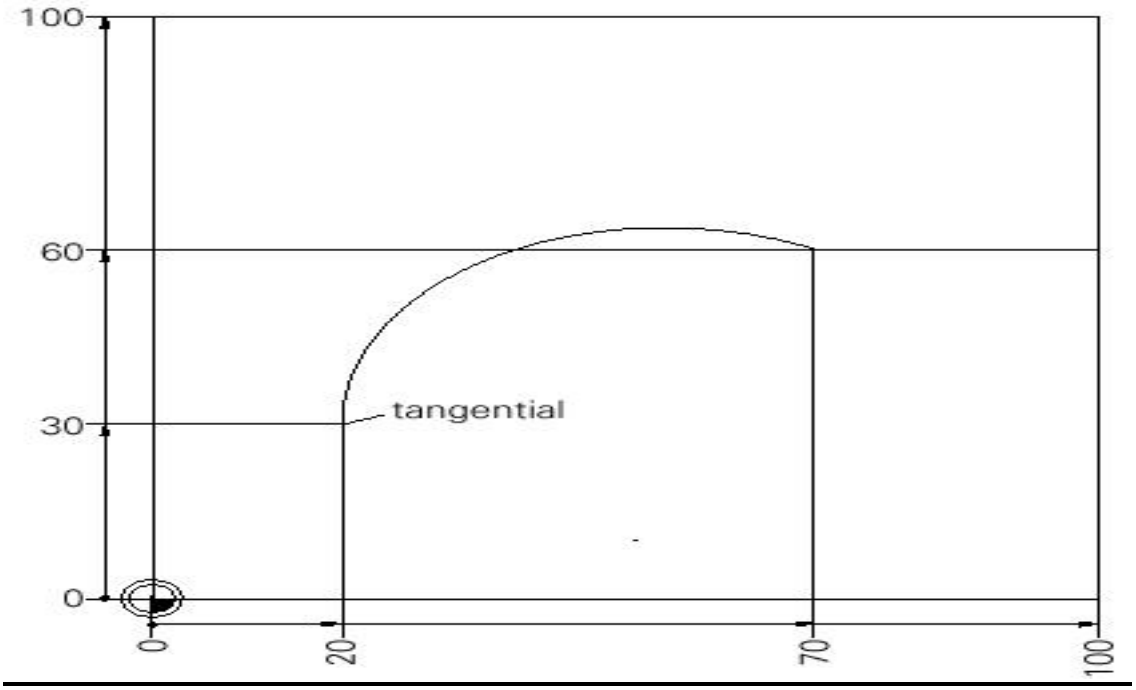


EDIT PROGRAM NO.: 811 (810) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	N/R
1	P	2	+40.000	+60.000	+0.000	+0.000	+0.000	+0.000
2	P	0	+31.000	+215.000	+0.000	+0.000	+0.000	+0.000
3	CW	0	+31.000	+35.000	+0.000	+0.000	+0.000	+31.000
4	CW	0	+19.000	+35.000	+0.000	+0.000	+0.000	+6.000
5	CCW	0	+19.000	+215.000	+0.000	+0.000	+0.000	+19.000
6	CW	0	+31.000	+215.000	+0.000	+0.000	+0.000	+6.000
7	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
8	P	2	+50.000	+25.000	+0.000	+0.000	+0.000	+0.000
9	P	2	+30.000	+200.000	+0.000	+0.000	+0.000	+0.000
10	P	0	+7.500	-70.000	+0.000	+0.000	+0.000	+0.000
11	CW	0	+7.500	+110.000	+0.000	+0.000	+0.000	+0.000
12	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
13	P	2	+50.000	+25.000	+0.000	+0.000	+0.000	+0.000
14	P	2	+30.000	+20.000	+0.000	+0.000	+0.000	+0.000
15	L	0	+7.500	+110.000	+0.000	+0.000	+0.000	+0.000
16	CW	0	+7.500	-70.000	+0.000	+0.000	+0.000	+0.000
17	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
18	P	2	+50.000	+25.000	+0.000	+0.000	+0.000	+0.000
19	P	2	+30.000	+200.000	+0.000	+0.000	+0.000	+0.000
E 20	L	0	+7.500	-70.000	+0.000	+0.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 812

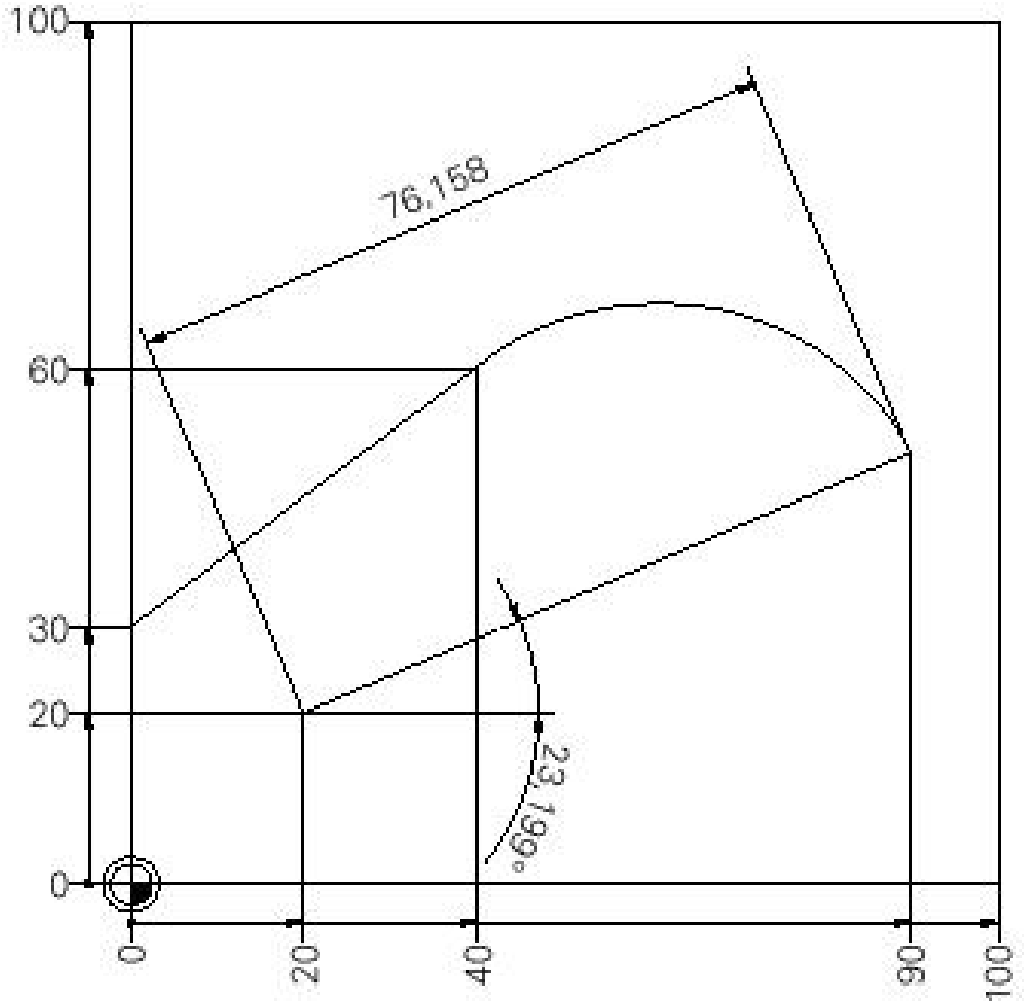


EDIT PROGRAM NO.: 812 (811) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	R+/C-
1	P	0	+20.000	+0.000	+0.000	+0.000	+0.000	+0.000
2	L	0	+20.000	+30.000	+0.000	+0.000	+0.000	+0.000
3	CW	11	+70.000	+60.000	+0.000	+0.000	+0.000	+0.000
4	L	0	+70.000	+0.000	+0.000	+0.000	+0.000	+0.000
E 5	L	0	+20.000	+0.000	+0.000	+0.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {> : QUIT

Program : 813

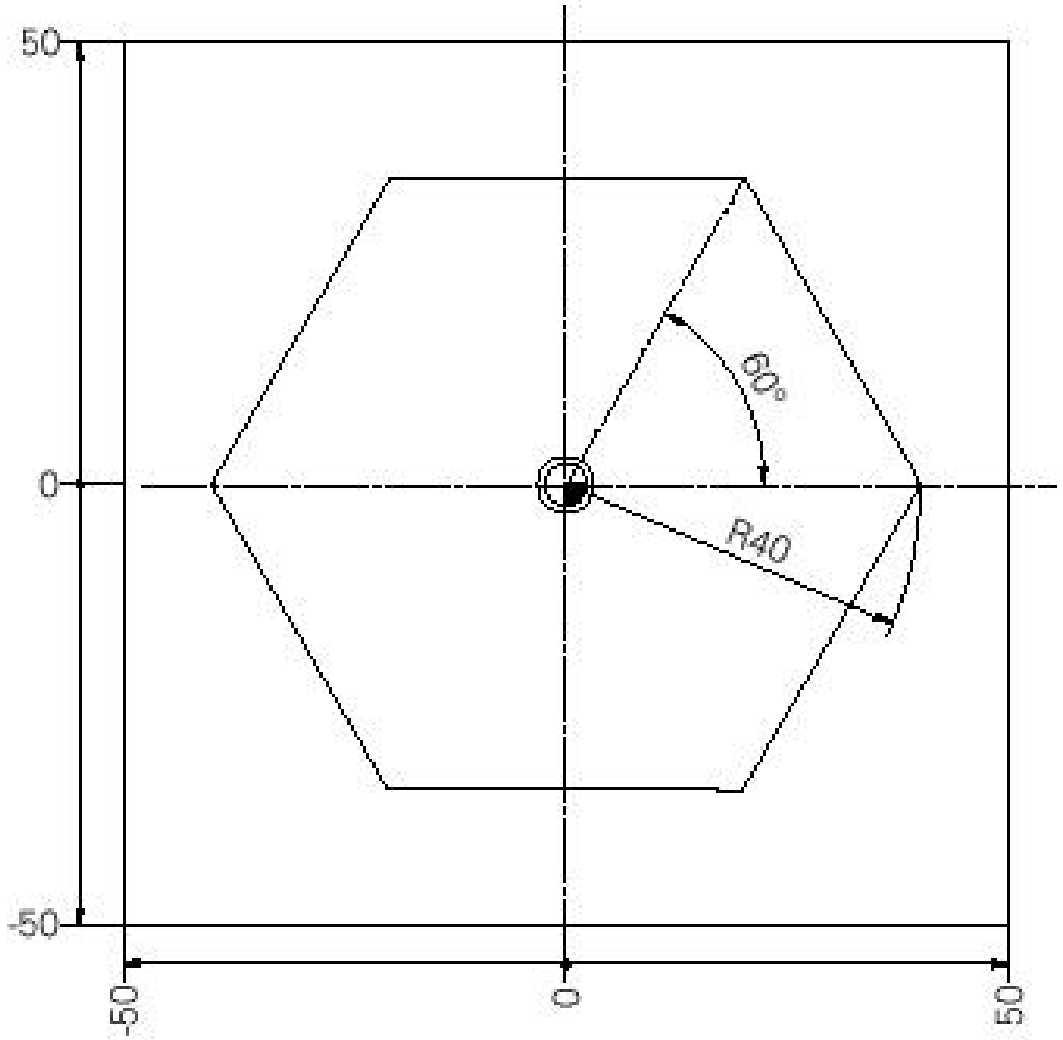


EDIT PROGRAM NO.: 813 (812) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	R+/C-
1	P	0	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
2	L	0	+90.000	+0.000	+0.000	+0.000	+0.000	+0.000
3	P	2	+20.000	+20.000	+0.000	+0.000	+0.000	+0.000
4	L	0	+76.158	+23.199	+0.000	+0.000	+0.000	+0.000
5	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
6	CCW	12	+40.000	+60.000	+0.000	+0.000	+0.000	+0.000
7	L	0	+0.000	+30.000	+0.000	+0.000	+0.000	+0.000
E 8	L	0	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 814

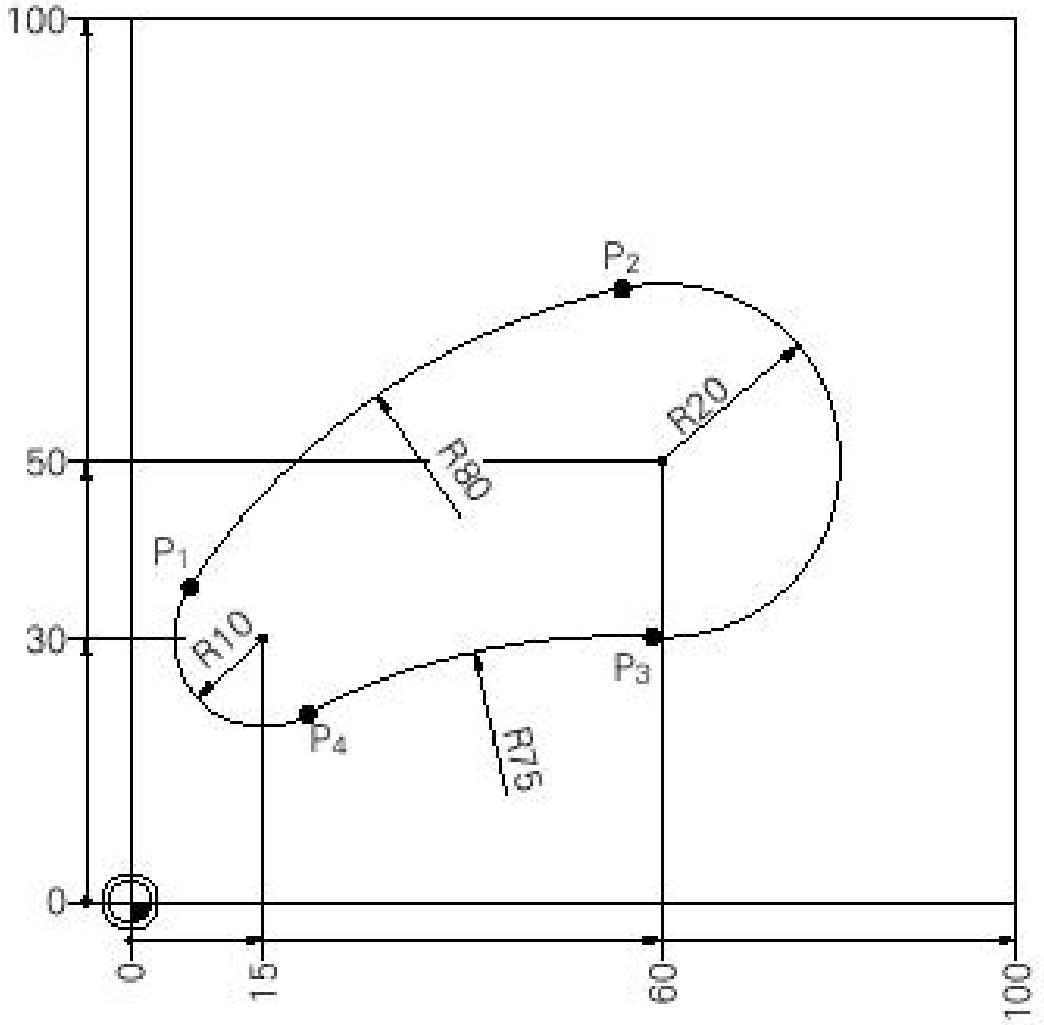


EDIT PROGRAM NO.: 814 (813) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	PR	PA	Z	N/R	N/R	R+/C-
1	P	2	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
2	P	0	+40.000	+0.000	+0.000	+0.000	+0.000	+0.000
3	L	0	+40.000	+60.000	+0.000	+0.000	+0.000	+0.000
4	L	0	+40.000	+120.000	+0.000	+0.000	+0.000	+0.000
5	L	0	+40.000	+180.000	+0.000	+0.000	+0.000	+0.000
6	L	0	+40.000	+240.000	+0.000	+0.000	+0.000	+0.000
7	L	0	+40.000	+300.000	+0.000	+0.000	+0.000	+0.000
E 8	L	0	+40.000	+0.000	+0.000	+0.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 815

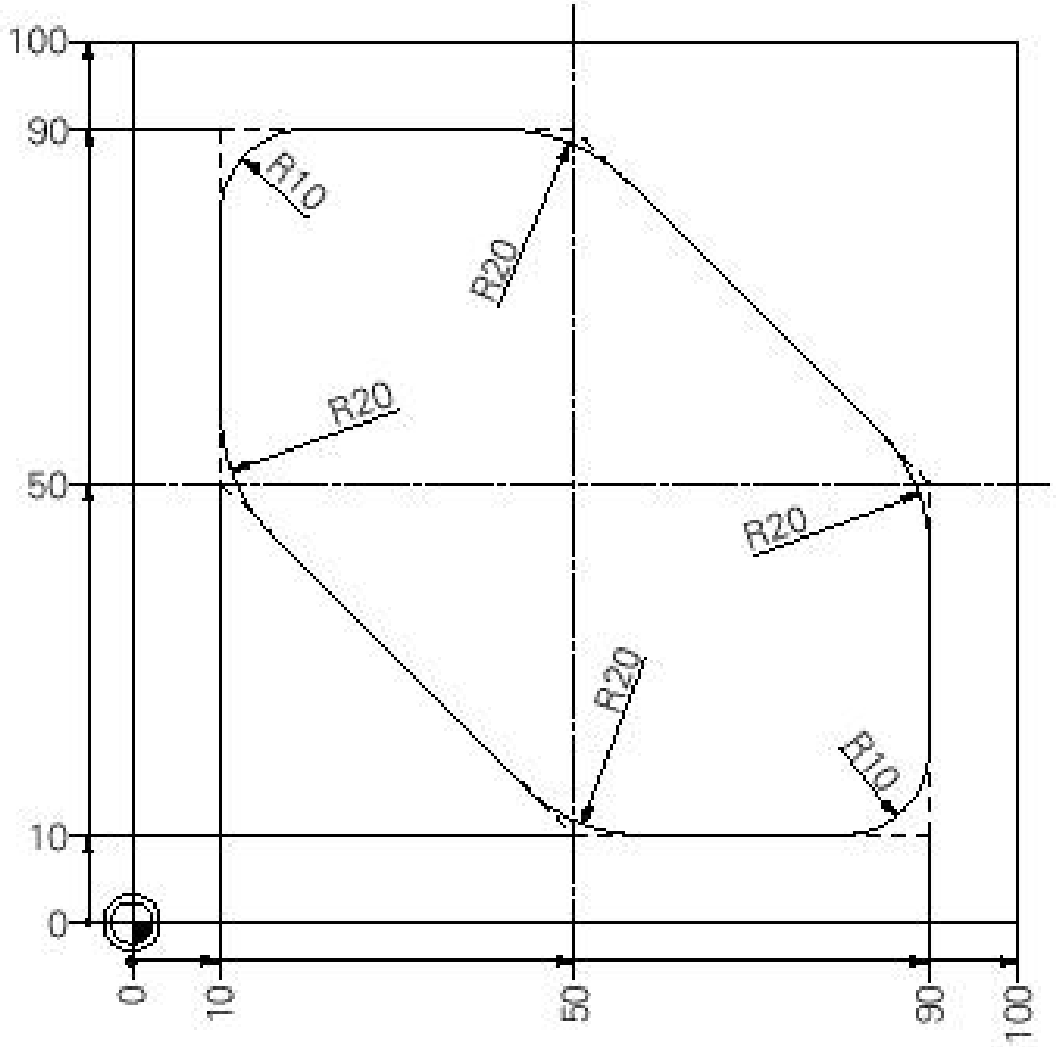


EDIT PROGRAM NO.: 815 (814) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	I	J	R+/C-
1	P	0	+5.000	+30.000	+0.000	+0.000	+0.000	+0.000
2	CW	10	+5.000	+30.000	+0.000	+10.000	+0.000	+0.000
3	CW	8	+0.000	+0.000	+0.000	+0.000	+0.000	+80.000
4	CW	10	+80.000	+50.000	+0.000	-20.000	+0.000	+0.000
5	CCW	8	+0.000	+0.000	+0.000	+0.000	+0.000	+75.000
E 6	CW	10	+5.000	+30.000	+0.000	+10.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 816

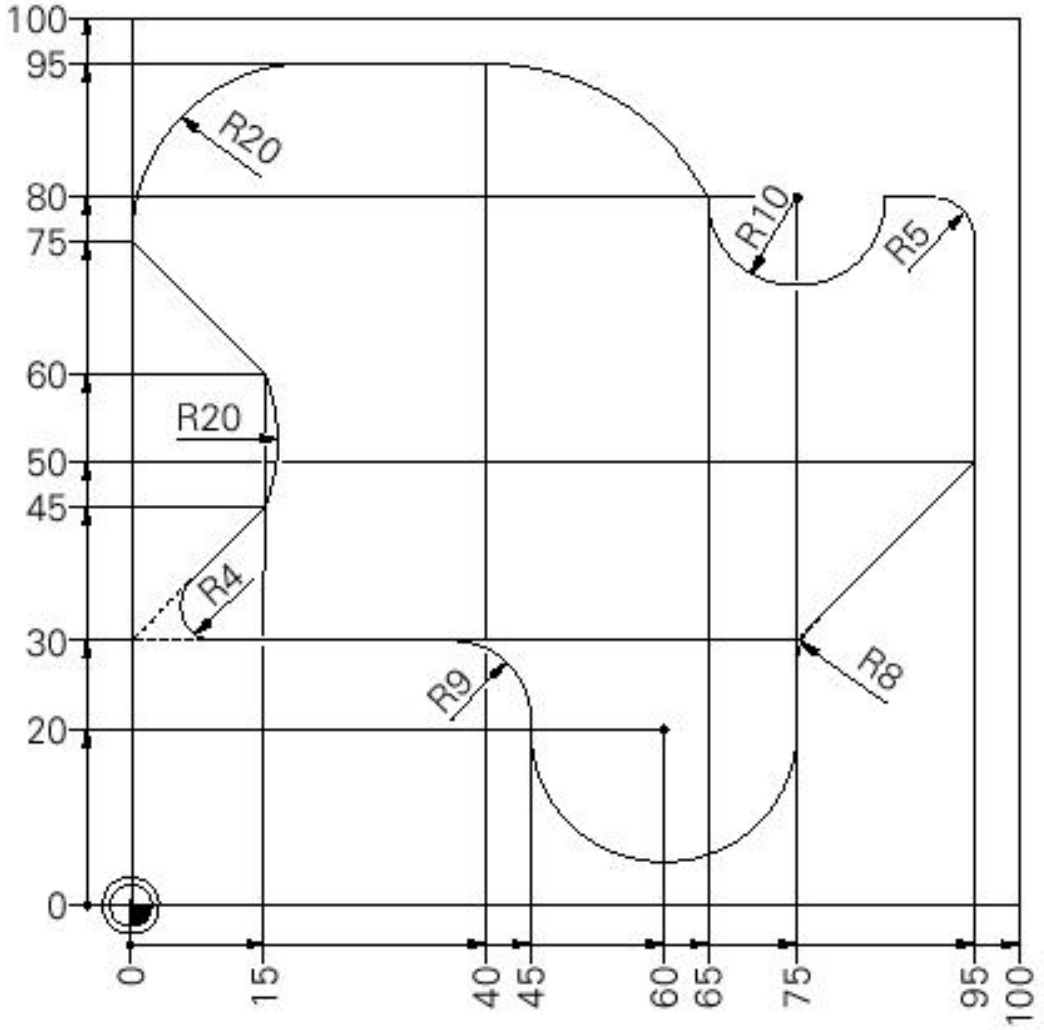


EDIT PROGRAM NO.: 816 (815) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	R+/C-
1	P	0	+10.000	+90.000	+0.000	+0.000	+0.000	+0.000
2	L	0	+50.000	+90.000	+0.000	+0.000	+0.000	+20.000
3	L	0	+90.000	+50.000	+0.000	+0.000	+0.000	+20.000
4	L	0	+90.000	+10.000	+0.000	+0.000	+0.000	+10.000
5	L	0	+50.000	+10.000	+0.000	+0.000	+0.000	+20.000
6	L	0	+10.000	+50.000	+0.000	+0.000	+0.000	+20.000
E 7	L	0	+10.000	+90.000	+0.000	+0.000	+0.000	+10.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 817

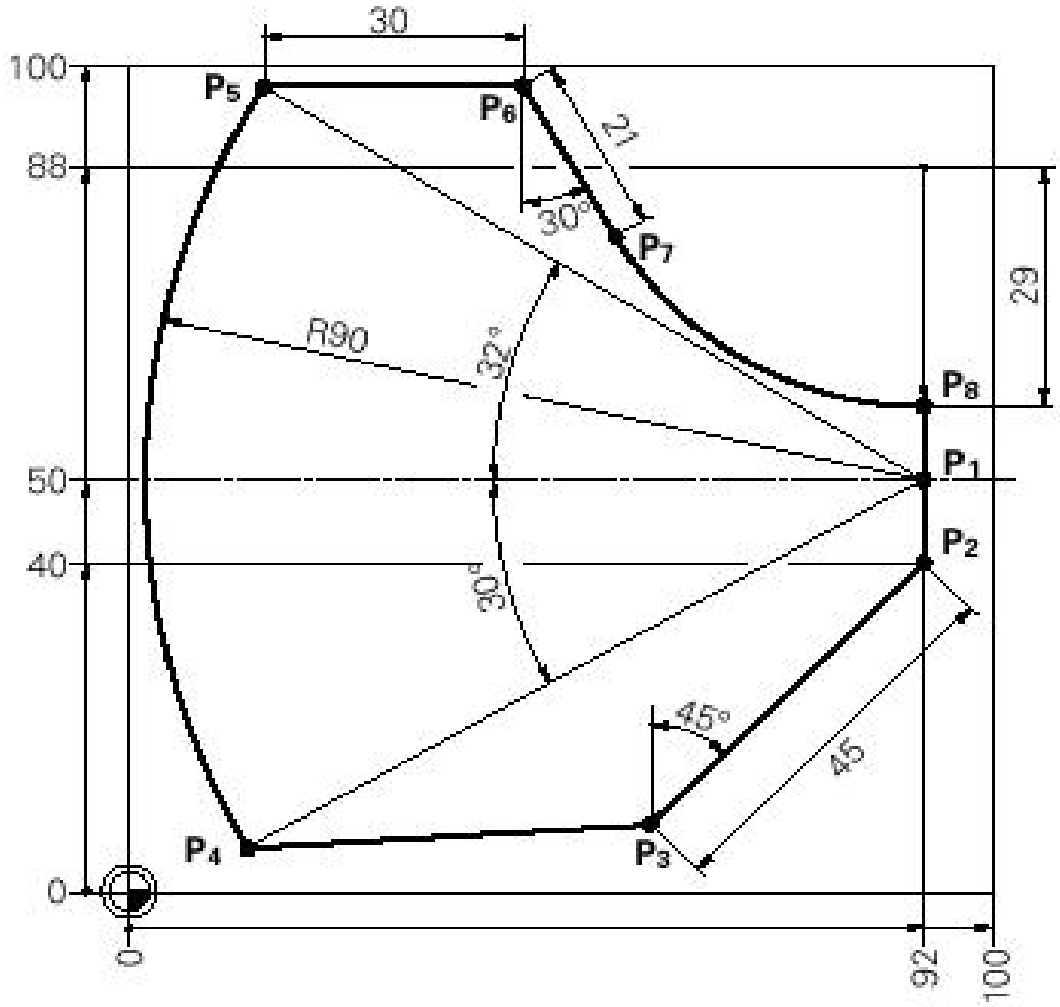


EDIT PROGRAM NO.: 817 (816) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	R+/C-
1	P	0	+0.000	+30.000	+0.000	+0.000	+0.000	+0.000
2	L	0	+15.000	+45.000	+0.000	+0.000	+0.000	+0.000
3	CCW	0	+15.000	+60.000	+0.000	+0.000	+0.000	+20.000
4	L	0	+0.000	+75.000	+0.000	+0.000	+0.000	+0.000
5	L	0	+0.000	+95.000	+0.000	+0.000	+0.000	+20.000
6	L	0	+40.000	+95.000	+0.000	+0.000	+0.000	+0.000
7	CW	0	+65.000	+80.000	+0.000	+0.000	+0.000	+25.000
8	CCW	0	+85.000	+80.000	+0.000	+10.000	+0.000	+0.000
9	L	0	+95.000	+80.000	+0.000	+0.000	+0.000	+5.000
10	L	0	+95.000	+50.000	+0.000	+0.000	+0.000	+0.000
11	L	0	+75.000	+30.000	+0.000	+0.000	+0.000	+8.000
12	L	0	+75.000	+20.000	+0.000	+0.000	+0.000	+0.000
13	CW	0	+45.000	+20.000	+0.000	-15.000	+0.000	+0.000
14	L	0	+45.000	+30.000	+0.000	+0.000	+0.000	+9.000
E 15	L	0	+0.000	+30.000	+0.000	+0.000	+0.000	+4.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 818

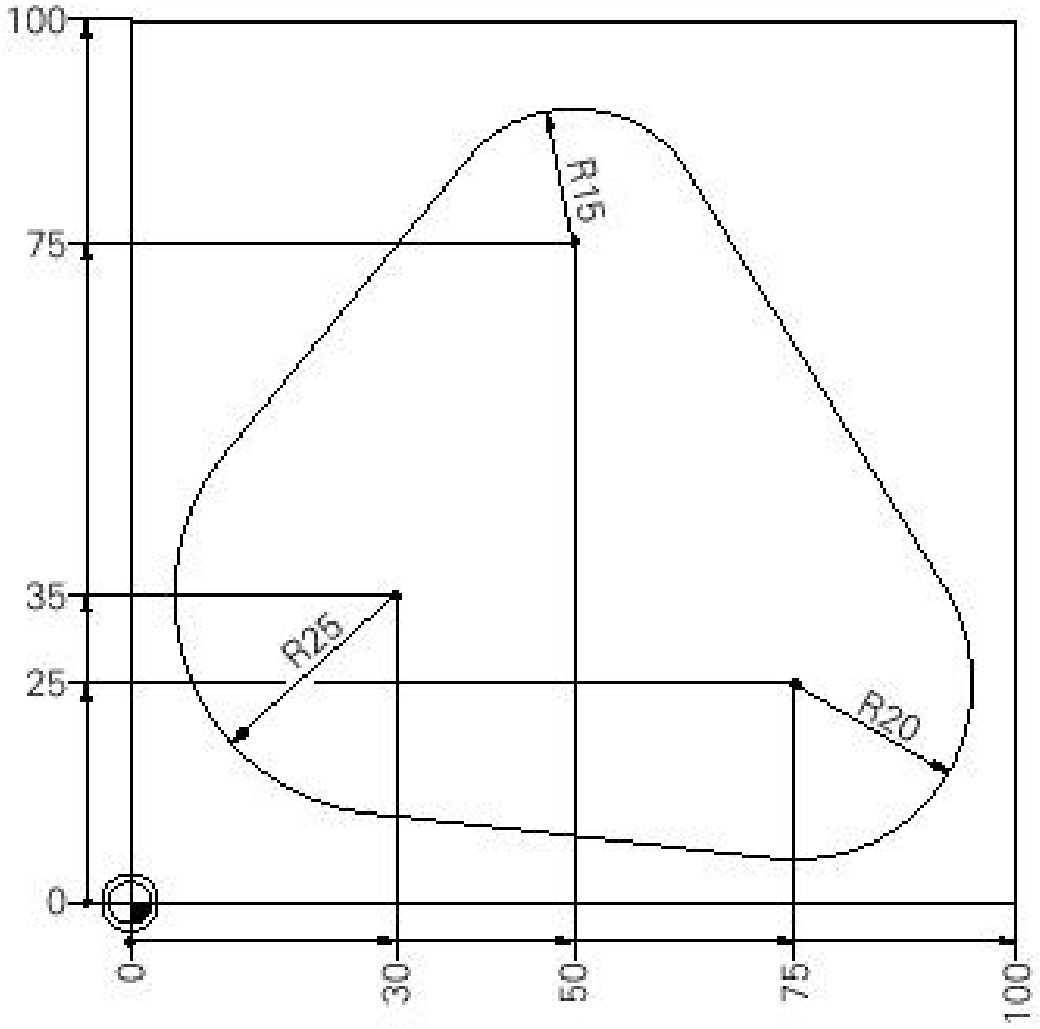


EDIT PROGRAM NO.: 818 (817) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	N/R
1	P	0	+92.000	+50.000	+0.000	+0.000	+0.000	+0.000
2	L	0	+92.000	+40.000	+0.000	+0.000	+0.000	+0.000
3	P	3	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
4	L	0	+45.000	+225.000	+0.000	+0.000	+0.000	+0.000
5	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
6	P	2	+92.000	+50.000	+0.000	+0.000	+0.000	+0.000
7	L	0	+90.000	+210.000	+0.000	+0.000	+0.000	+0.000
8	CW	0	+90.000	+148.000	+0.000	+0.000	+0.000	+90.000
9	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
10	P	3	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
11	L	0	+30.000	+0.000	+0.000	+0.000	+0.000	+0.000
12	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
13	P	3	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
14	L	0	+21.000	-60.000	+0.000	+0.000	+0.000	+0.000
15	P	4	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
16	CCW	11	+92.000	+59.000	+0.000	+0.000	+0.000	+0.000
E 17	L	0	+92.000	+50.000	+0.000	+0.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

Program : 819

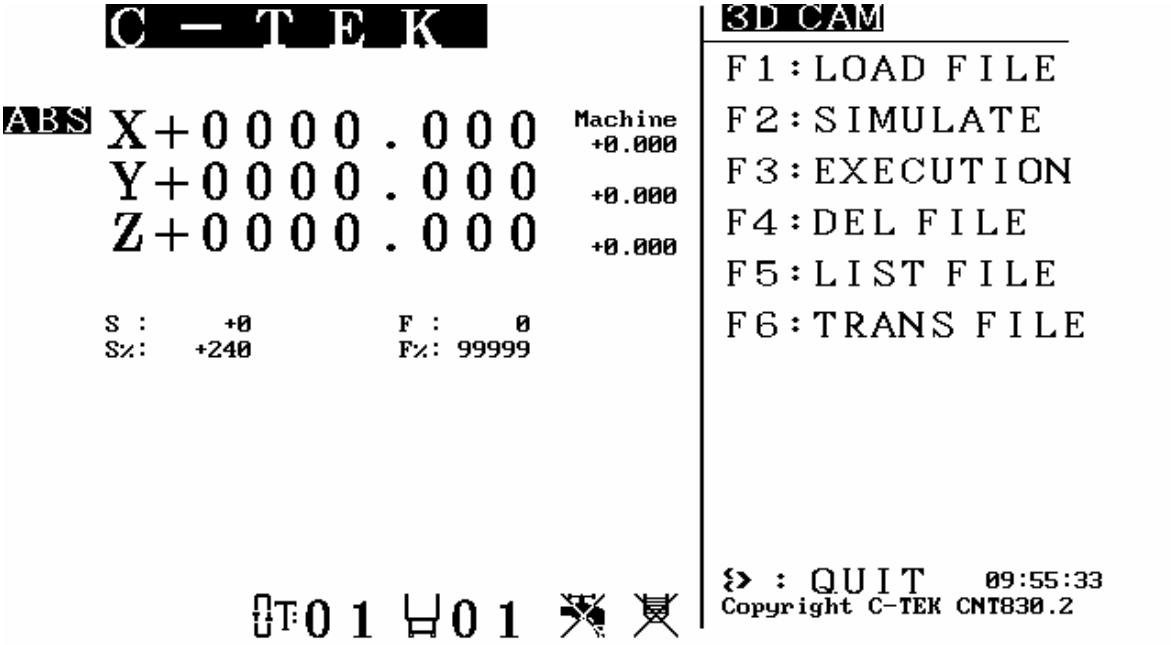


EDIT PROGRAM NO.: 819 (818) Coordinate System : 0 (ABS=0,INC=1)

Step	Cmd	Fn	X	Y	Z	N/R	N/R	N/R
1	P	0	+5.000	+35.000	+0.000	+0.000	+0.000	+0.000
2	CW	10	+5.000	+35.000	+0.000	+25.000	+0.000	+0.000
3	L	8	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
4	CW	10	+50.000	+90.000	+0.000	+0.000	-15.000	+0.000
5	L	8	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
6	CW	10	+95.000	+25.000	+0.000	-20.000	+0.000	+0.000
7	L	8	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
E 8	CW	10	+5.000	+35.000	+0.000	+25.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:DATA/END F6:CHK/DRAW {>:QUIT

13. 3D CAM



Characteristics:

1. **Fast connection:** User can directly load the 3D CAD/CAM NC file from the floppy without connecting through the network. Or use the transferred file from network connecting.
2. **Path emulation:** User can simulate the cutting path before it really goes to machine the workpiece to prevent inadequate cutting situation.
3. **The 3D transferred file can be revised again.**

F1: LOAD FILE

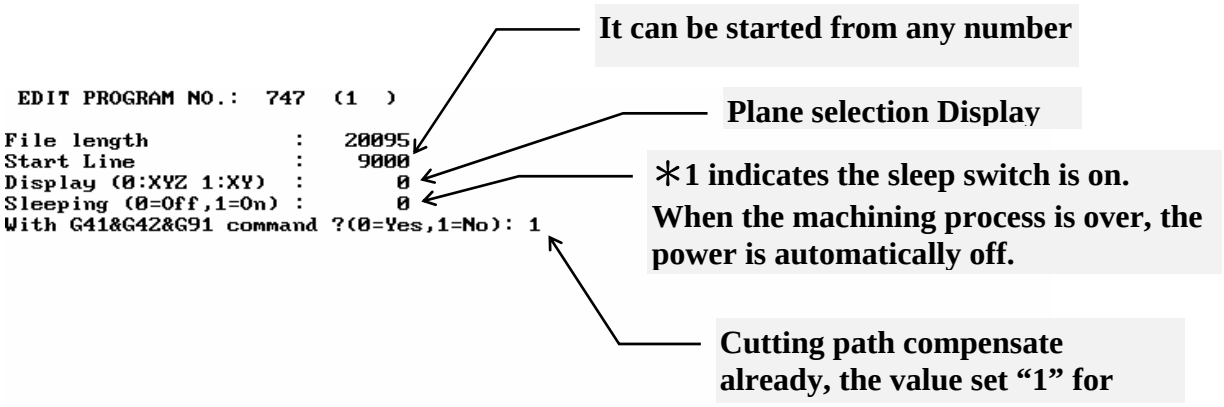
Description:

1. Input the file no. 747（or 0 to 999).

EDIT PROGRAM NO.: 747 (1) LINE NO.: 1
TOTAL LINE : 20095

CMD							
	+5000.00	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
M03	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G90	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
0Z	+0.000	+0.000	+50.000	+0.000	+0.000	+0.000	+0.000
G90	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G80	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G00	+0.000	+0.000	+50.000	+0.000	+0.000	+0.000	+0.000
G00	+60.021	-54.021	+50.000	+0.000	+0.000	+0.000	+0.000
G00	+60.021	-54.021	+15.000	+0.000	+0.000	+0.000	+0.000
G00	+60.021	-54.021	+3.789	+0.000	+0.000	+0.000	+0.000
F	+500.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G01	+60.021	-54.021	-2.211	+0.000	+0.000	+0.000	+0.000
F	+900.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G01	+60.021	+54.021	-2.211	+0.000	+0.000	+0.000	+0.000
G01	-60.021	+54.021	-2.211	+0.000	+0.000	+0.000	+0.000

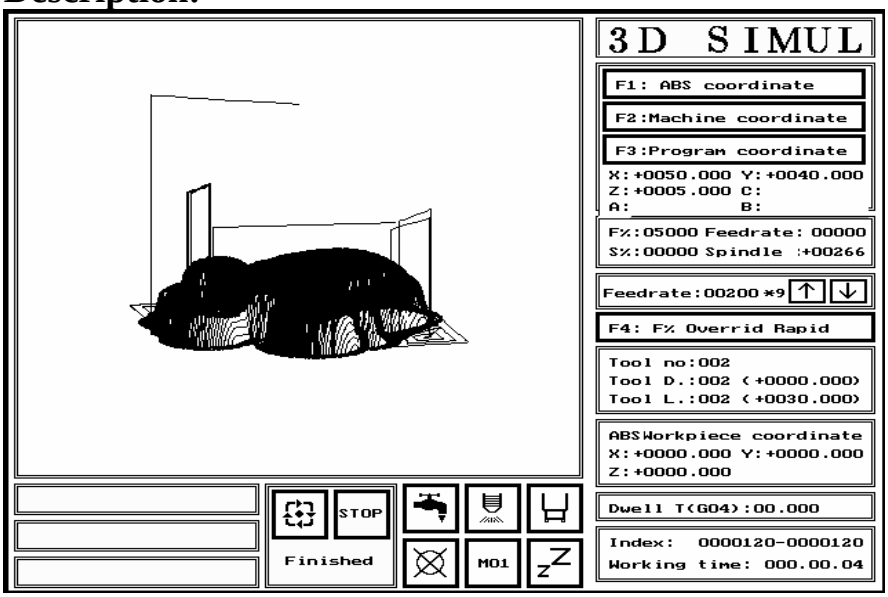
F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:GO LINE {> : QUIT


EDIT PROGRAM NO.: 747 (1)
File length : 20095
Start Line : 9000
Display (0:XYZ 1:XY) : 0
Sleeping (0=Off,1=On) : 0
With G41&G42&G91 command ?(0=Yes,1=No): 1
It can be started from any number
Plane selection Display
∗1 indicates the sleep switch is on. When the machining process is over, the power is automatically off.
Cutting path compensate already, the value set “1” for

∗Press the QUIT key to go back 3D CAM screen.

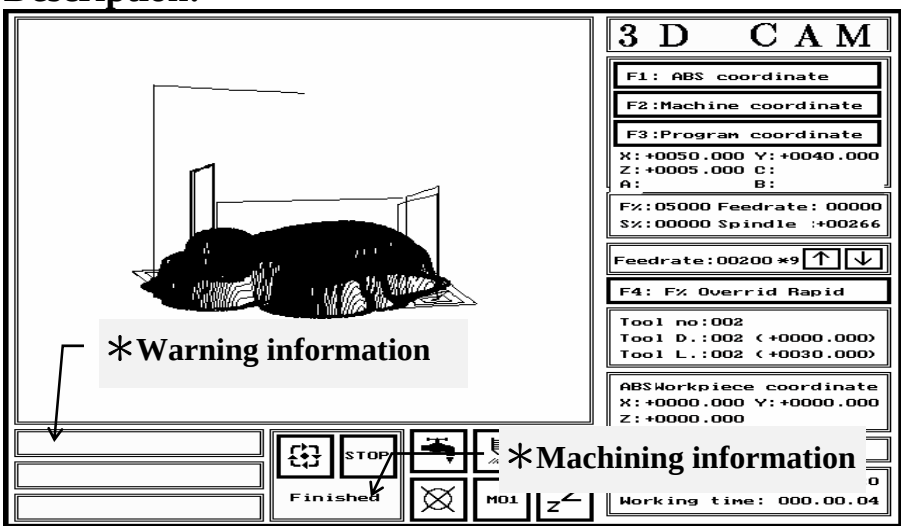
F2: DRAW

Description:



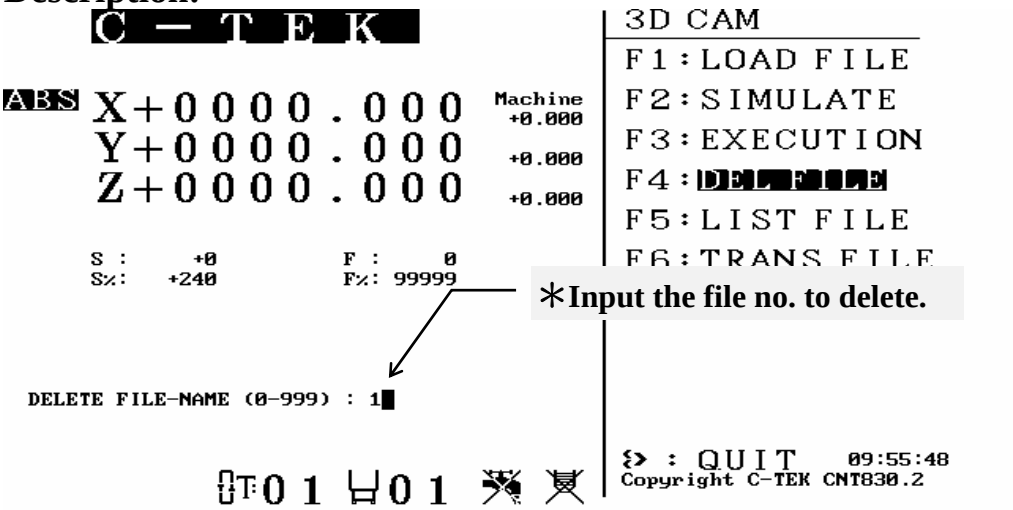
F3: EXECUTION

Description:



F4: DEL FILE

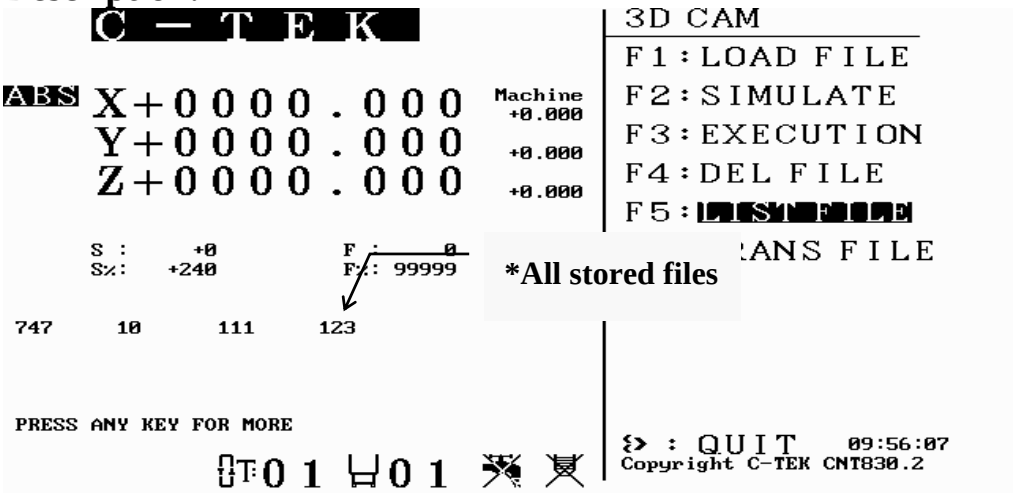
Description:



＊Input the file no. to delete.

F5: LIST FILE

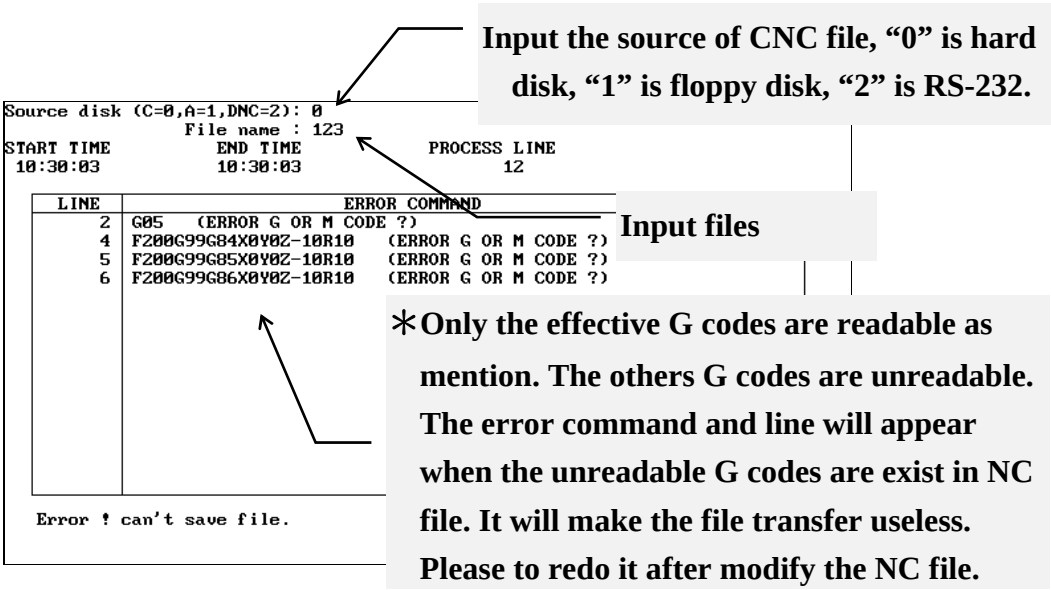
Description:



*All stored files

F6: TRANS FILE

Description:



Input the source of CNC file, “0” is hard disk, “1” is floppy disk, “2” is RS-232.

Input files

＊Only the effective G codes are readable as mention. The others G codes are unreadable. The error command and line will appear when the unreadable G codes are exist in NC file. It will make the file transfer useless. Please to redo it after modify the NC file.

*** Effective G code:**

G00, G01, G02, G03, G15, G17, G18, G19, G21, G40, G41, G42, G43, G44, G49, G54, G55, G56, G57, G58, G59, G64, G69, G73, G74, G76, G80, G81, G82, G83, G84, G85, G86, G87, G88, G89, G90, G92, G98, G99

*** Effective M code:**

M00, M01, M02, M03, M04, M05, M06, M07, M08, M09, M30.

*** Caution;**

1. **The command form of tool change: TnM6 or TnM06**
   Tn is tool number. When n=0, the screen shows "Error message"

2. **The command form of rpm: SnM3 or SnM03**
   Sn is the revolution speed.

3. **The command form of opposite rpm: SnM4 or SnM04**
   Sn is the revolution speed.

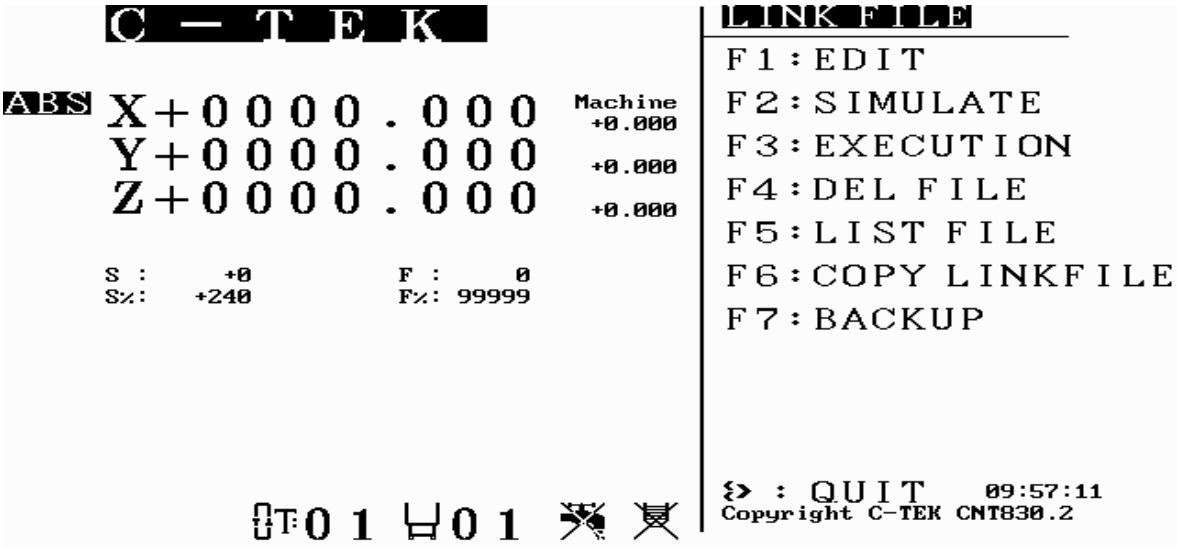
4. **M00, M01, M02: Performed as suspending order**

5. **Transferred with RS232,『%』 must be added in the head and the end of program**

Standard RS232 transferred contract:

BAUDRATE: 9600
DATABIT: 8
STOPBIT: 1
PARITY: NONE

14.LINK FILE

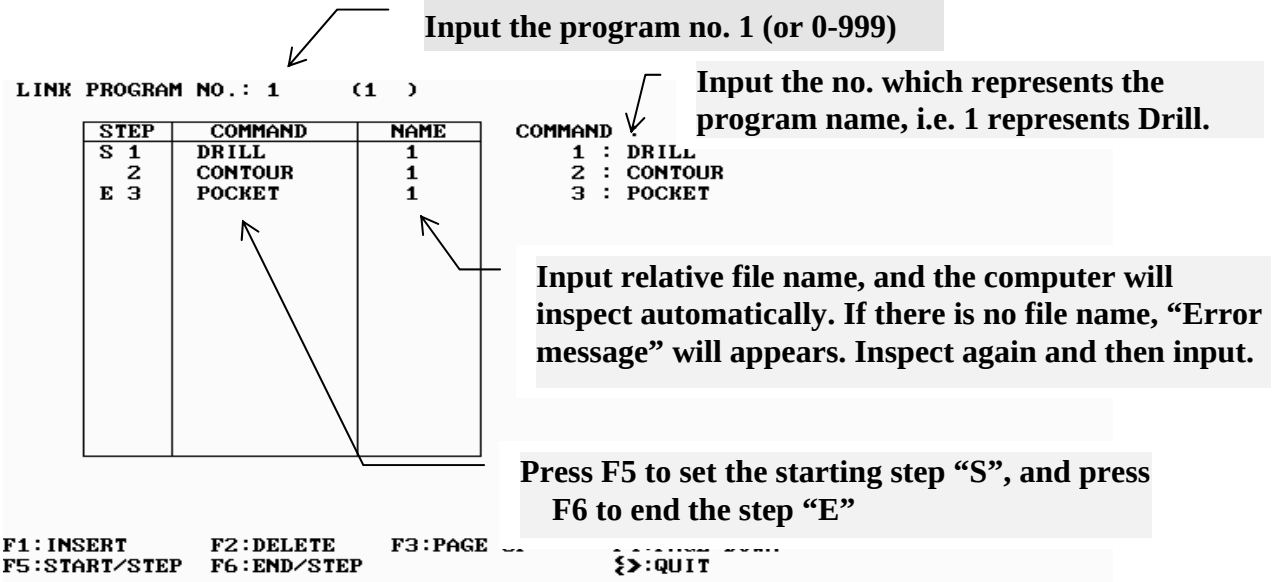


Characteristics:

1. Link file: You can link up all the edited files of DRILL, CONTOUR, POCKET, TAP, BORE and execute them at the same time.
During execution meet TOOL CHANGE command, the system which has TOOL CHANGE function will change the tool immediately.
If the system has no TOOL CHANGE function, the program will suspend for a while until it completes changing tool manual and press the button on the remote control box of" CONTINUOUS RUNNING." , the machining operation is continued.

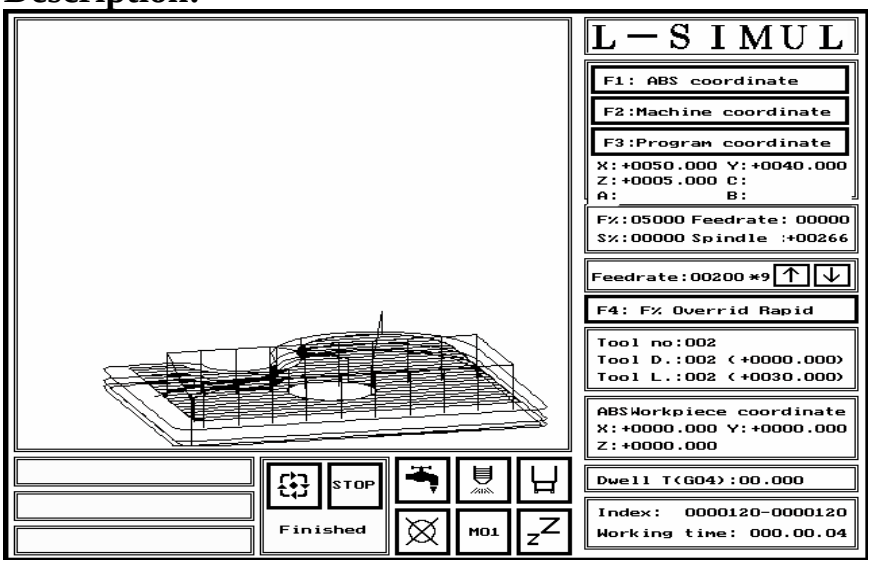
F1: EDIT:

Description:

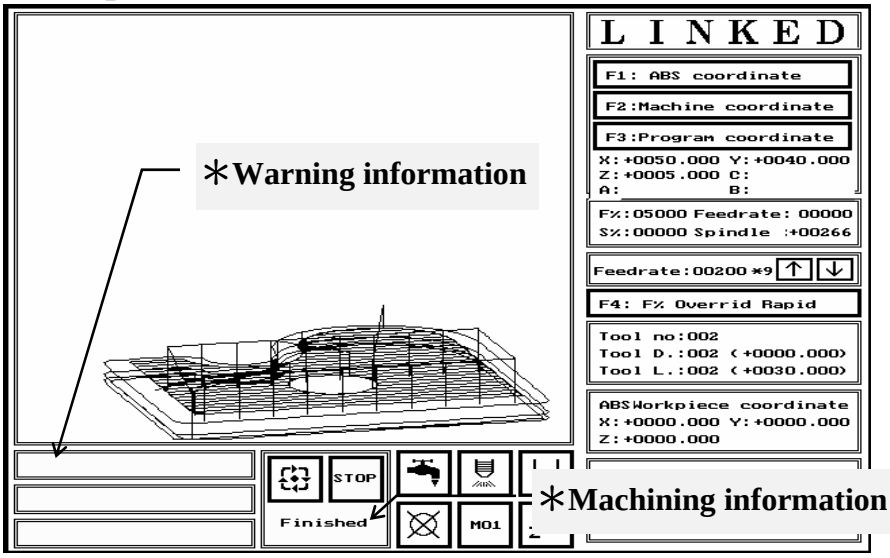

Input the program no. 1 (or 0-999)
LINK PROGRAM NO.: 1 (1)
STEP COMMAND NAME
S 1 DRILL 1
2 CONTOUR 1
E 3 POCKET 1
COMMAND :
1 : DRILL
2 : CONTOUR
3 : POCKET
Input the no. which represents the program name, i.e. 1 represents Drill.
Input relative file name, and the computer will inspect automatically. If there is no file name, "Error message" will appears. Inspect again and then input.
Press F5 to set the starting step "S", and press F6 to end the step "E"
F1:INSERT F2:DELETE F3:PAGE
F5:START/STEP F6:END/STEP
{>:QUIT

- **P**ress "quit" key, the computer will inspect automatically the file link existent, if there is any wrong file or not exist, it can not be save and display error line, after the "Save file" show up , press "N" to quit. After checking, If everything is right, save files directly and goes back to "FILE LINK".

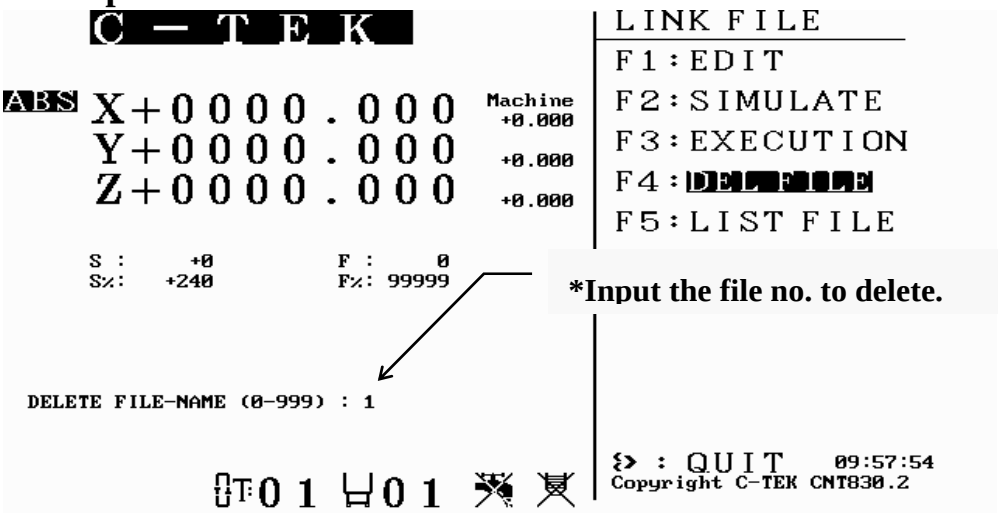
F2: DRAW
Description:



F3: EXECUTION
Description:

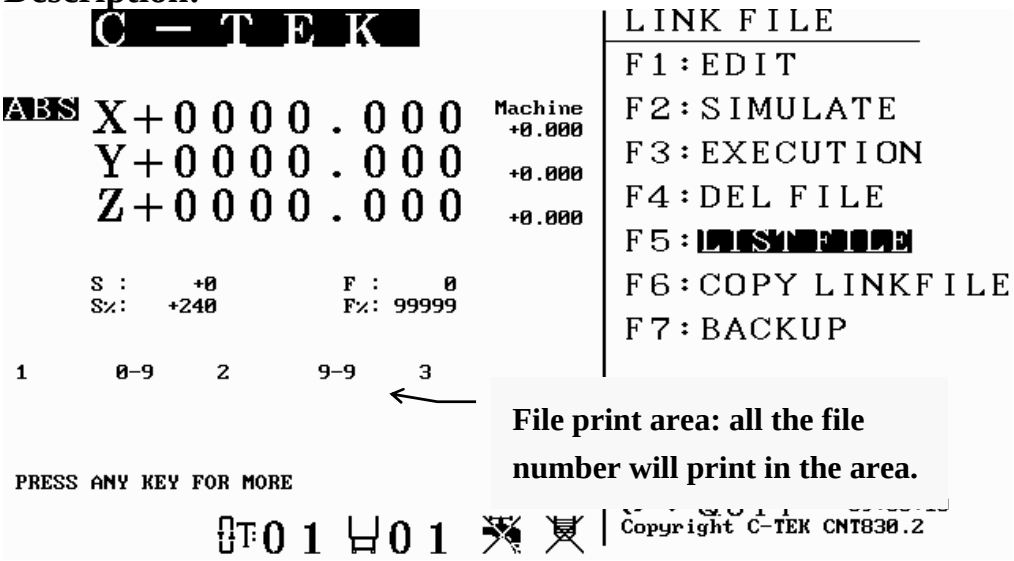


F4: DEL FILE:
Description:



F5: LIST FILE

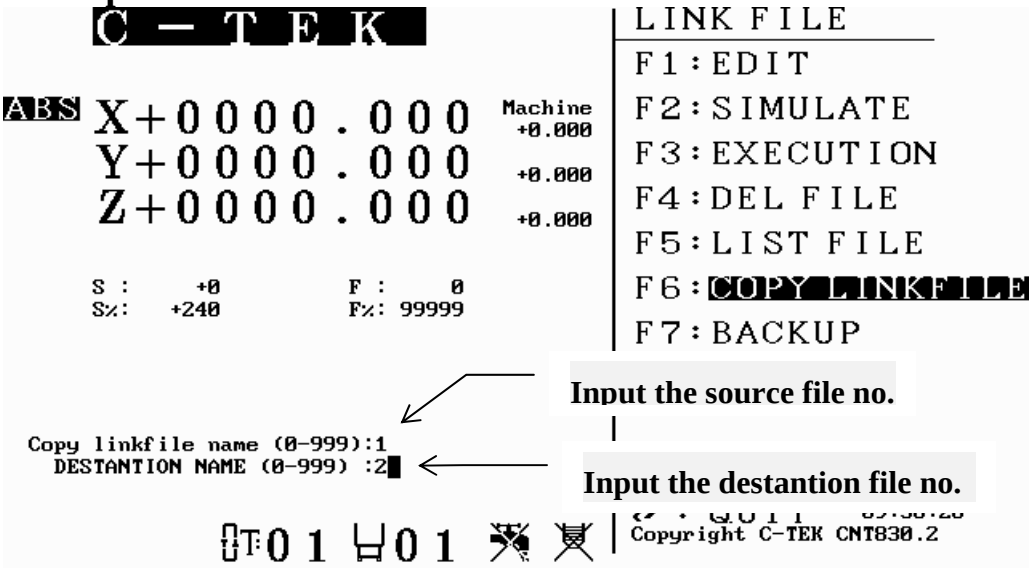
Description:



File print area: all the file number will print in the area.

F6: COPY LINK FILE

Description:

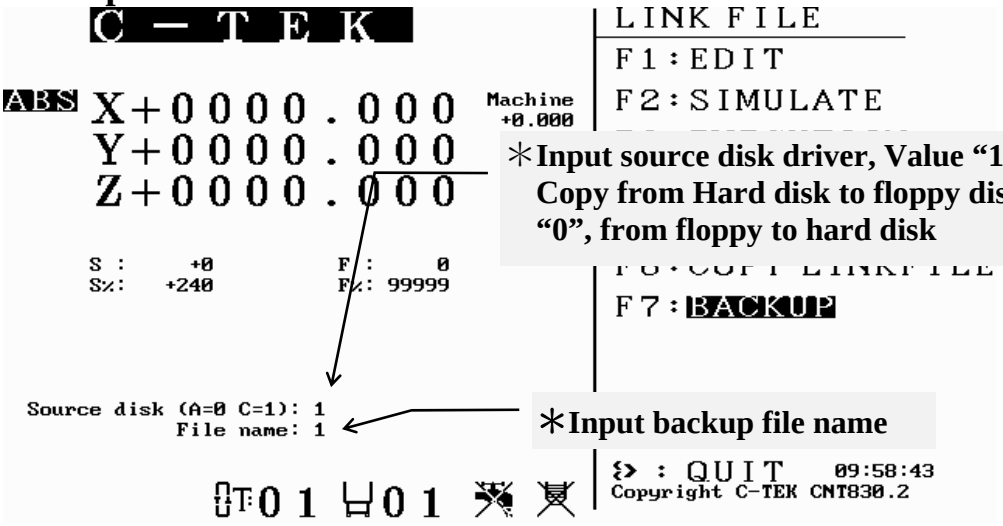


Input the source file no.

Input the destantion file no.

F7: BACKUP

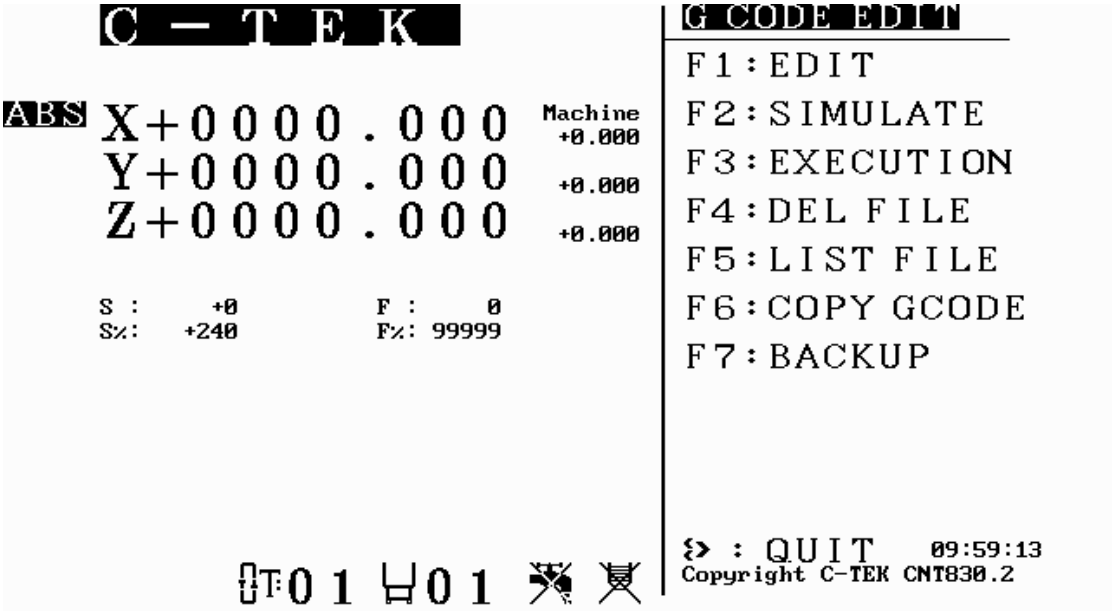
Description:



＊Input source disk driver, Value “1” Copy from Hard disk to floppy disk “0”, from floppy to hard disk

＊Input backup file name

15. G CODE EDIT



Characteristics:

With G Code to edit the program**.**

F1: EDIT:

Description:

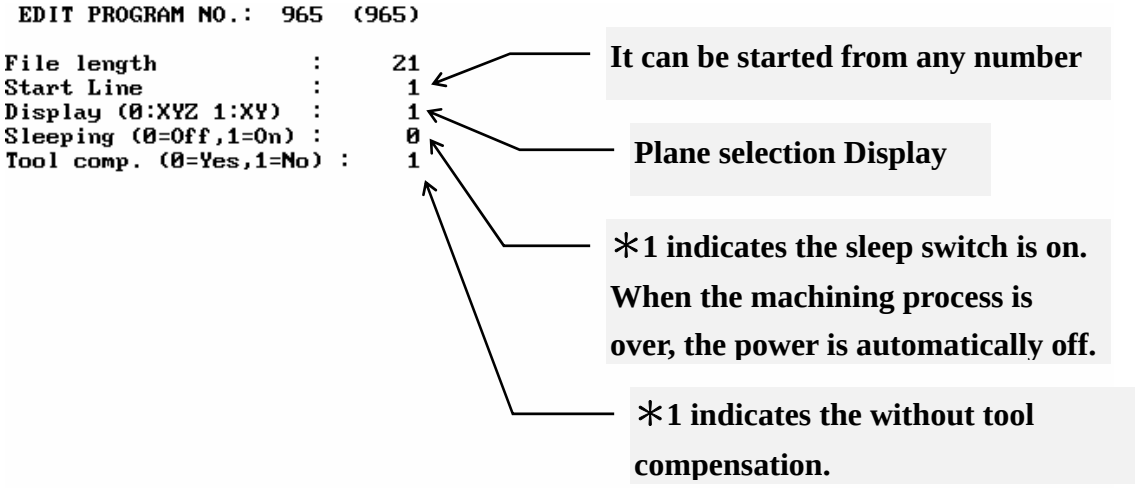
1. Input the program no. 1 (or 0-999)

EDIT PROGRAM NO.: 1 (1) LINE NO.: 1
TOTAL LINE : 21

CMD							
G54	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G00	-10.000	-100.000	+0.000	+0.000	+0.000	+0.000	+0.000
G00	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
M98	1	1	+0.000	+0.000	+0.000	+0.000	+0.000
G00	+150.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
M98	1	1	+0.000	+0.000	+0.000	+0.000	+0.000
G00	+150.000	+150.000	+0.000	+0.000	+0.000	+0.000	+0.000
M98	1	1	+0.000	+0.000	+0.000	+0.000	+0.000
G00	+0.000	+150.000	+0.000	+0.000	+0.000	+0.000	+0.000
M98	1	1	+0.000	+0.000	+0.000	+0.000	+0.000
M30	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
N	1	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G91	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G01	+100.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G01	+0.000	+75.000	+0.000	+0.000	+0.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:GO LINE F6:HELP {> : QUIT

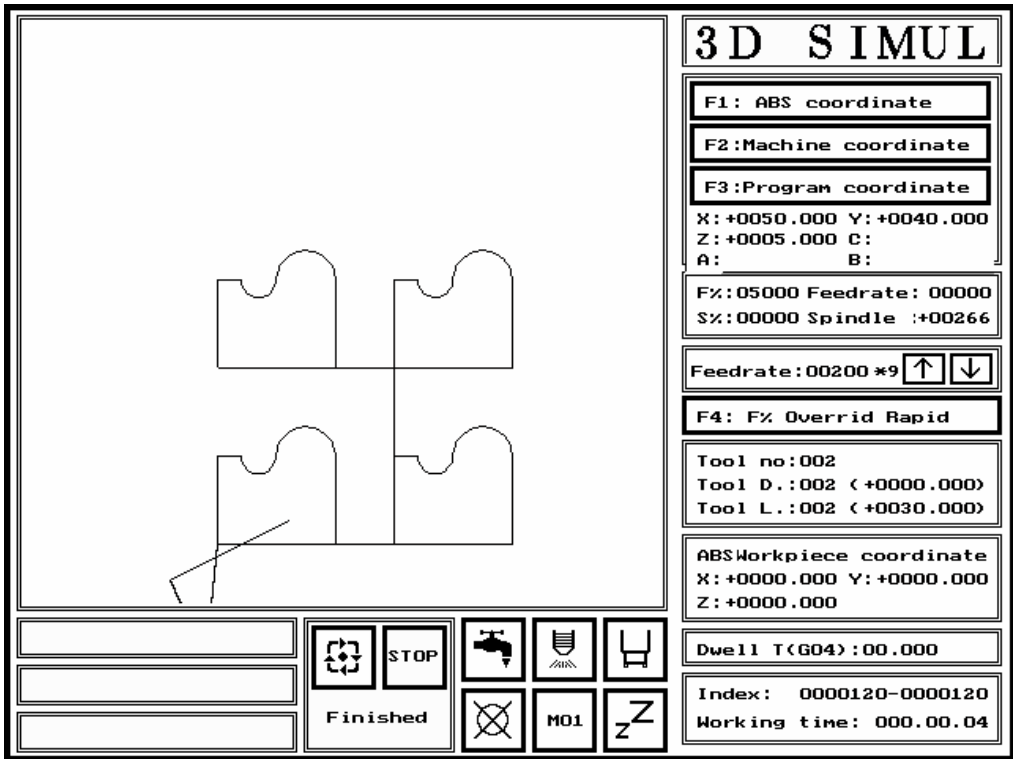
Parameter setting :


EDIT PROGRAM NO.: 965 (965)
File length : 21
Start Line : 1
Display (0:XYZ 1:XY) : 1
Sleeping (0=Off,1=On) : 0
Tool comp. (0=Yes,1=No) : 1
It can be started from any number
Plane selection Display
＊1 indicates the sleep switch is on. When the machining process is over, the power is automatically off.
＊1 indicates the without tool compensation.

＊Press the QUIT key to back to G CODE EDIT screen.

F2: DRAW

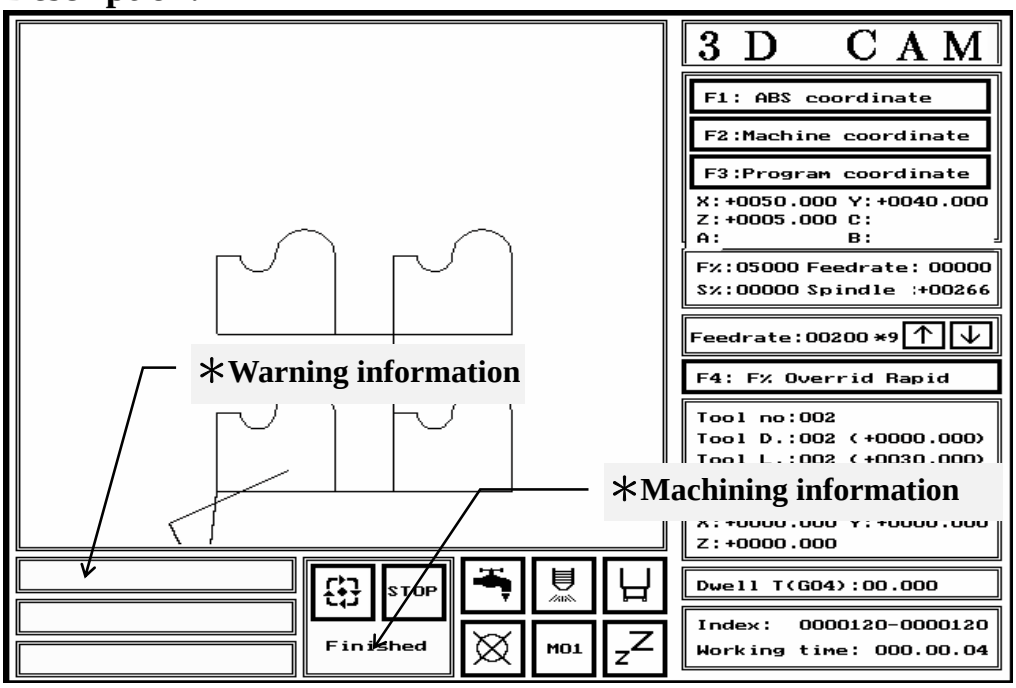
Description:



Press “QUIT” to back to G CODE EDIT screen

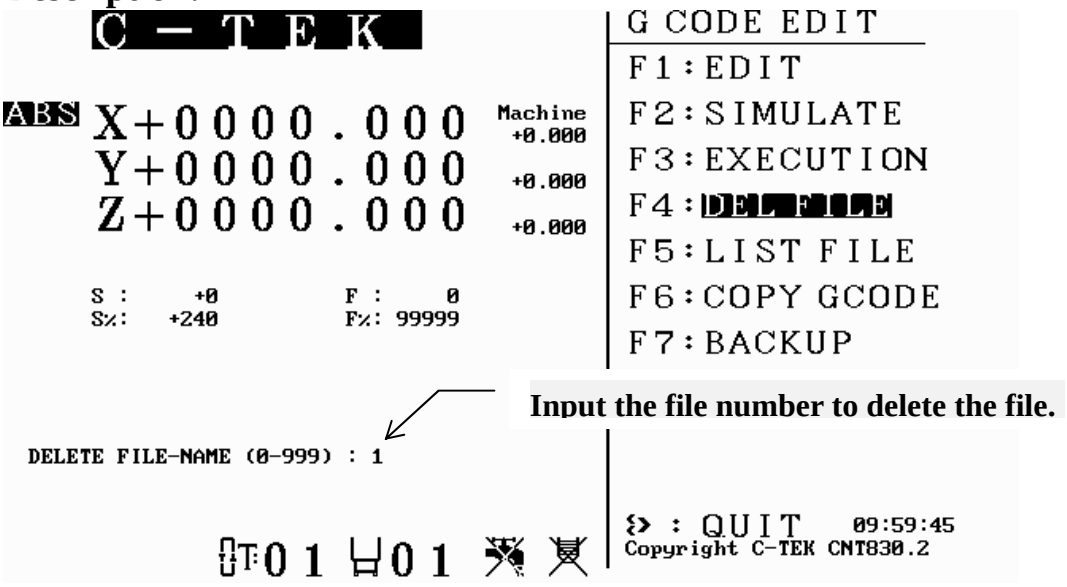
F3: EXECUTION

Description:



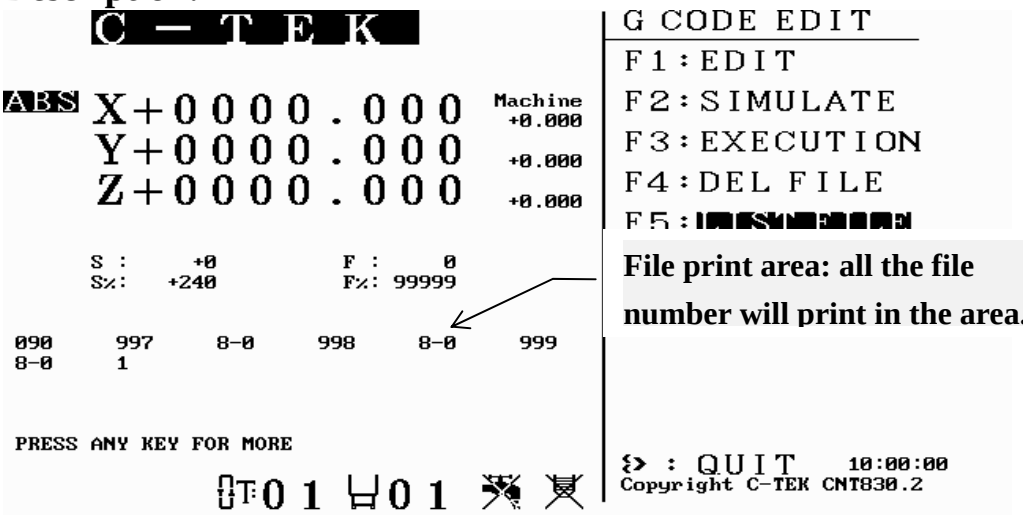
F4: DEL FILE:

Description:



F5: LIST FILE

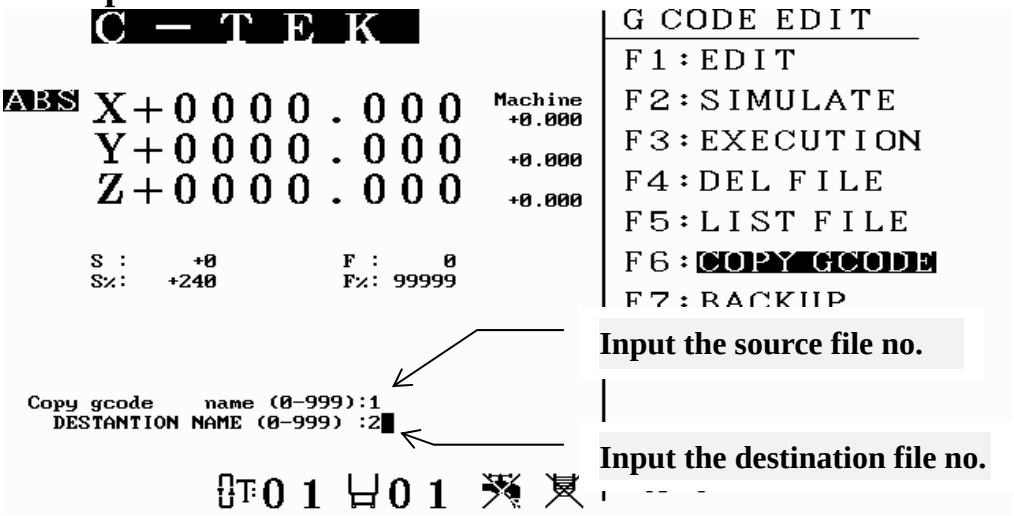
Description:



File print area: all the file number will print in the area.

F6: COPY GCODE

Description:

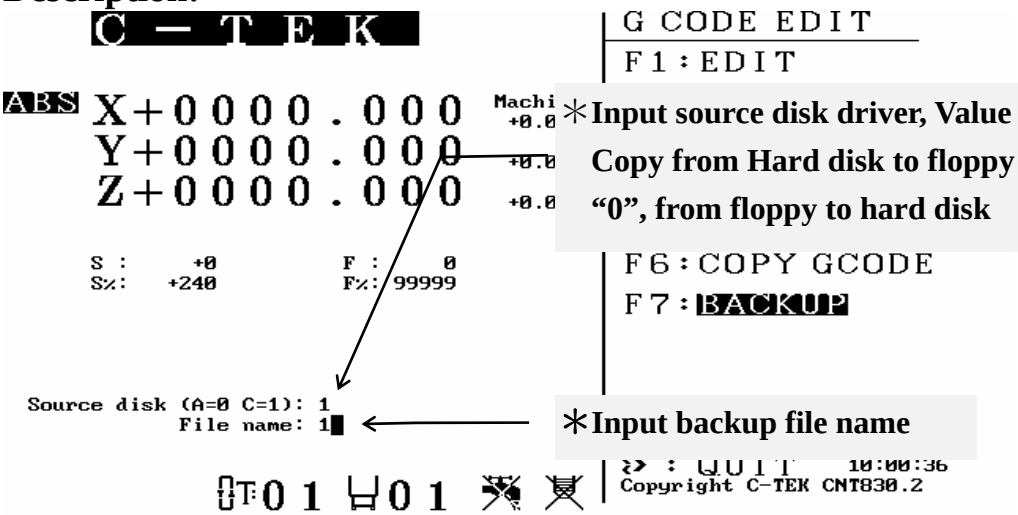


Input the source file no.

Input the destination file no.

F7: BACKUP

Description:



＊Input source disk driver, Value “1” Copy from Hard disk to floppy disk “0”, from floppy to hard disk

＊Input backup file name

＊Effective G Code:

G CODE	FORM	FUNCTION
G00	G00X____Y____Z____	Positioning
G01	G01X____Y____Z____F____	Linear Interpolation
G02	G17{G02}X__Y__{R__ }F__ {G03} {I__J__}	Circular interpolation (clockwise)
G03	G18{G02}Z__X__{R__ }F__ {G03} {K__I__} G19{G02}Y__Z__{R__ }F__ {G03} {J__K-_}	Circular interpolation (counterclockwise)
G04	G04X____	Dwell
G15	G15	Polar coordinate system cancel
G16	G16	Polar coordinate system set
G17	G17	Plane setting X－Y
G18	G18	Plane setting X－Z
G19	G19	Plane setting Y－Z
G21	G21	Metric system setting
G40	G40	Cutter radius compensation cancel
G41	G41D__	Cutter radius compensation left
G42	G42D__	Cutter radius compensation right
G43	G43H__	Tool length offset(+)
G44	G44H__	Tool Length offset(-)
G49	G49	Tool Length compensation cancel
G54	G54	Work coordinate system 1 setting
G55	G55	Work coordinate system 2setting
G56	G56	Work coordinate system 3 setting
G57	G57	Work coordinate system 4 setting
G58	G58	Work coordinate system 5 setting
G59	G59	Work coordinate system 6 setting
G64	G64	Exact Stop check cancel
G69	G69	Coordinate rotating cancel
G73	G73 X__Y__Z__ R__Q__	High speed canned cycle
G74	G74 X__Y__Z__ R	Left Tap canned cycle
G76	G76 X__Y__Z__ P__Q__	Fixed cycle cancel
G80	G80	Constant cycle cancel
G81	G81 X__Y__Z__ R__	Drilling canned cycle
G82	G82 X__Y__Z__ R__P__	Deep hole canned cycle
G83	G83 X__Y__Z__ R__ Q__	Canned cycle

G84	G84 X__Y__Z__ R__	Rigid Tap cycle
G85	G85 X__Y__Z__ R__	Bore cycle
G86	G86 X__Y__Z__ R	Bore cycle
G87	G87 X__Y__Z__ R__ P__Q__	Back Bore cycle
G88	G88 X__Y__Z__ R__ P__	Bore cycle
G89	G89 X__Y__Z__ R__ P__	Bore cycle
G90	G90	Absolute value coordinate system
G92	G92	Machine coordinate system setting
G98	G98	Initial Point return constant cycle
G99	G99	R point return constant cycle

M CODE	**FORM**	**FUNCTION**
M00	M00	Program suspend
M01	M01	Program suspend
M02	M02	Program suspend
M03	S___M03	Spindle rotation
M04	S___M04	Spindle reverse rotation
M05	M05	Spindle Stop
M06	T__M06	Tool Change
M07	M07	Air on
M08	M08	Coolant on
M09	M09	Air state and coolant off
M20	M20G04X____	C axis unlock (brake off) and dwell **(MTIO.pcb CR11)**
M21	M21	Input signal for MTIO.pcb (J4-7,8)
M22	M22G04X____	C axis lock (brake on) and dwell **(MTIO.pcb CR11)**
M24	M24G04X____	B axis unlock (brake off) and dwell **(MTIO.pcb CR12)**
M25	M25	Input signal for MTIO.pcb (J4-5,6)
M26	M26G04X____	B axis lock (brake on) and dwell **(MTIO.pcb CR12)**
M28	M28	Input signal for MTIO.pcb (J4-3,4)
M30	M30	Program stop
M98	M98P__L__	Subroutine P__& Times L ___
M99	M99	Subroutine stop
N	N___	Subroutine Code

✳Special call for M98 9995-9999 sample :

EDIT PROGRAM NO.: 1 (1) LINE NO.: 1
TOTAL LINE : 13

CMD	X	Y	Z				
G00	+0.000	+0.000	+50.000	+0.000	+0.000	+0.000	+0.000
G00	+0.000	+0.000	+10.000	+0.000	+0.000	+0.000	+0.000
M98	9995	1	+0.000	+0.000	+0.000	+0.000	+0.000
M98	9996	1	+0.000	+0.000	+0.000	+0.000	+0.000
M98	9997	1	+0.000	+0.000	+0.000	+0.000	+0.000
M98	9998	1	+0.000	+0.000	+0.000	+0.000	+0.000
M98	9999	1	+0.000	+0.000	+0.000	+0.000	+0.000
M98	1	2	+0.000	+0.000	+0.000	+0.000	+0.000
M30	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
N	1	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
F	+500.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
G01	+100.000	+0.000	+10.000	+0.000	+0.000	+0.000	+0.000
M99	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000

F1:INS F2:DEL F3:PG/UP F4:PG/DN F5:GO LINE F6:HELP {>:QUIT

M98 9995 1 : Call DRILL module program file number 1.

M98 9996 1 : Call TAP module program file number 1.

M98 9997 1 : Call BORE module program file number 1.

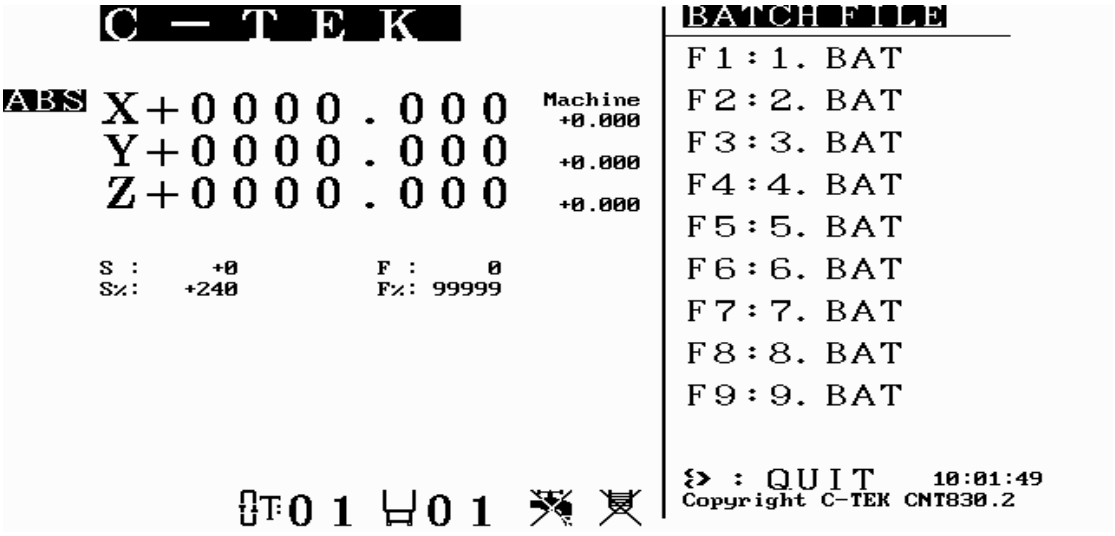
M98 9998 1 : Call CONTOUR module program file number 1.

M98 9999 1 : Call POCKET module program file number 1.

M98 1 2 : Normal call subroutine file number 1 for two times.

If program number is smaller than 9995. It's a normal call. It must have has a subroutine N mapping.

16. BATHCH FILE



FUNCTION:

F1: 1.BAT
F2: 2.BAT
F3: 3.BAT
F4: 4.BAT
F5: 5.BAT
F6: 6.BAT
F7: 7.BAT
F8: 8.BAT
F9: 9.BAT

→ Provide nine function of Batch. User can write the commands in each of the nine files through edition program (in C:\CTEK>). Just press F1-F9 to execute it.

For Example:

1. User has a CNC file to machining, filename is 1.CNC. It is compress into floppy disk and named **1.ABC**. User want to use F2: 2.BAT to expand it in the controller:

2. BAT: CD\DAT — Go into DAT directory

Copy A: 1.ABC — Copy 1.ABC from A: Drive

ABC E 1 — Use ABC file to expand 1.ABC into 1.cnc.

CD\CTEK — Go Back to CTEK Directory.

2. User want to use the outward editor, such as PE2 or EDIT to edit CNC, just copy the outward editor into the hard disk c:\ctek directory and write 3.BAT as follow:

PE2

(or EDIT)

Default batch file :

F1 : 1.BAT Use for network connect and download file from server.

```
CD\TCPIP
CALL FTP.BAT
CD\CTEK
```

F9 : 9.BAT Use for delete all CNC files.

```
DEL C:\DAT\*.CNC
DEL C:\DAT\3DCA*.*
DEL C:\DAT\3DPAR*.*
```

APPENDIX 一. SYSTEM PARAMETER

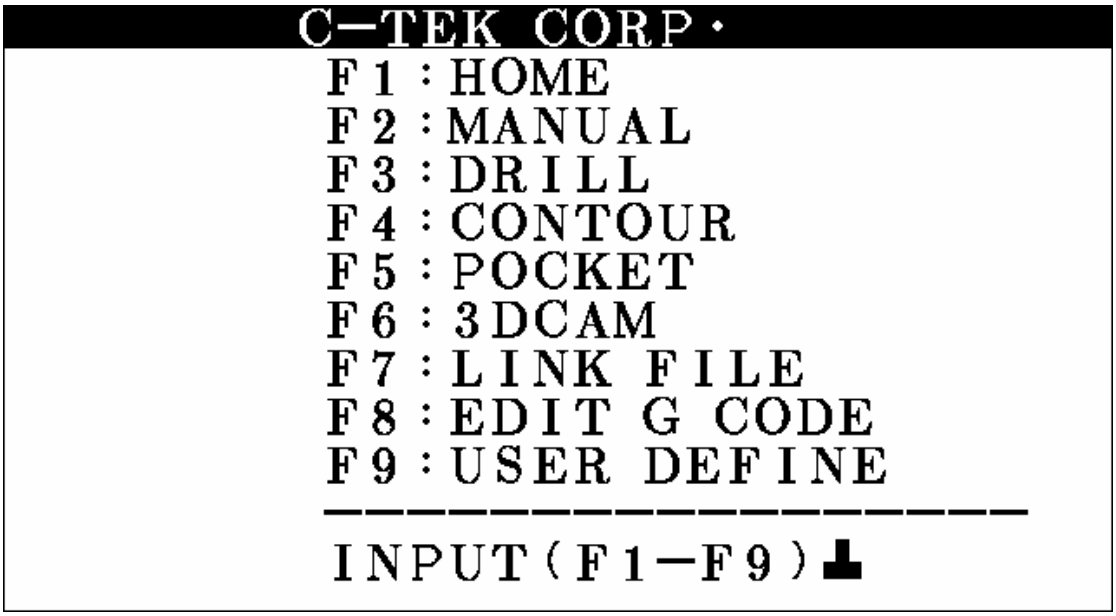


FIG. 1 **MAIN FUNCTION**

INTO SYSTEM:

1. From MAIN FUNCTION (FIG. 1) Press "⟪" and "1" will into INPUT PASSWORD (FIG. 2)

2. **Press** "⟪ " and "2" will into DOS system.

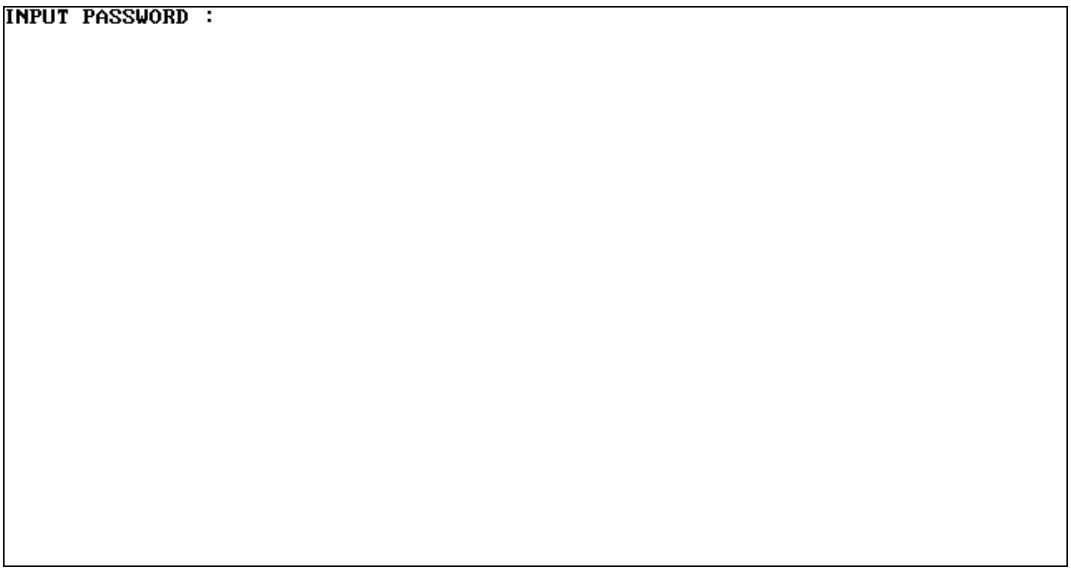


FIG. 2: INPUT PASSWORD

Input password "530509" (this code not show up) into select (FIG. 3)

```
SYSTEM PARAMETER MODIFY
-----------------------
 1: System
 2: Travel error compensation
 3: XYZ axes set
 4: MPG set
 5: S axis
 6: Motor set static gain
 7: Motor set dynamic gain
 8: C axis
 9: B axis
10: Miscellaneous
11: DNC set
12: I/O test
13: Tune
14: Turret ATC1 parameter
15: Motor parameter
16: Quit
----------------------------
SELECT (1-16) ?
```

FIG. 3 SELECT

3. Press "1" into system parameter (Fig. 4)

```
 System parameter
Language (0-99)          :      1 (CHINESE)
Metric(0)/Inch(1)        :      0
X backlash (um)          :     15
Y backlash (um)          :     24
Z backlash (um)          :     37
Sample time (ms)         :      2
Hi resolution            :     10
Switch debouns           :      5
G54-59 Ref.(0=Abs,1=Mac):      0
Display Refresh          :     30
Default 3DCAM G41-2 comp:      1
```

FIG. 4 SYSTEM PARAMETER

4. Move the cursor to item for modifing.

Ex. (1) move the cursor to Language item.

(2) Press 0 ↵ (input complete)

(3) Press "⏩" to quit and back to main function.

Parameter Description:

Language: 「0」English , 「1」Chinese, 「2」 Dutch,「3」Italian,「4」German 「5」Spanish,「6」Finland,「7」Hungarian

Metric : 「0」Mertic , 「1」Inch

X Backlash : X-axis backlash offset , (uM)

Y Backlash : Y-axis backlash offset , (uM)

Z Backlash : Z-axis backlash offset , (uM)

Sample time: Stand value "2"

Hi resolution : Standard value "10"

Switch debouns : Standard value " 5", I/O reading time control.

G54-59 Ref.(0=Abs 1=Mac): Value"0" means G54-59 coordinate is relative to the zero point of workpiece. Value"1" Means G54-59 coordinate is relative to machine zero point.

Display refresh: Rate of scan display, Standard value "10"

Default 3dcam G41-2 comp: CNC file conversion , the value add to parameter automatically, set value"1" for saving input time when loading from 3dcam.

5. Press "2" into the pitch error compensation (Fig. 5)

PITCH X(m/m) = 20 PITCH Y(m/m) = 30 PITCH Z(m/m) = 40
Unit : (um) F1:X SET F2:Y SET F3:Z SET F4:XYZ SET

No.	+X	-X	+Y	-Y	+Z	-Z
1	+1	+0	+0	+0	+0	+1
2	+2	+0	+0	+0	+0	+2
3	+3	+0	+0	+0	+0	+3
4	+4	+0	+0	+0	+0	+4
5	+5	+0	+0	+0	+0	+5
6	+6	+0	+0	+0	+0	+6
7	+7	+0	+0	+0	+0	+7
8	+8	+0	+0	+0	+0	+8
9	+9	+0	+0	+0	+0	+9
10	+0	+0	+0	+0	+0	+0
11	+1	+0	+0	+0	+0	+1
12	+2	+0	+0	+0	+0	+2
13	+3	+0	+0	+0	+0	+3
14	+4	+0	+0	+0	+0	+4
15	+5	+0	+0	+0	+0	+5
16	+6	+0	+0	+0	+0	+6
17	+7	+0	+0	+0	+0	+7
18	+8	+0	+0	+0	+0	+8
19	+9	+0	+0	+0	+0	+9
20	+0	+0	+0	+0	+0	+0

FIG. 5 **Pitch Error Compensation**

Description:

M. Pitch X (m/m): X-axis offset pitch. Ex. Set "50", means per 50m/m for offset.

M. Pitch Y (m/m): Y-axis offset pitch. Ex. Set "50", means per 50m/m for offset.

M. Pitch Z (m/m): Z-axis offset pitch. Ex. Set "50", means per 50m/m for offset.

No. : Sequence of compensation.

+X, +Y, +Z: Home returned, XYZ axes forward to positive direction **Pitch error compensation value for 50 point.**

-X, -Y, -Z: Home returned, XYZ axes forward to negative direction **Pitch error compensation value for 50 point.**

Ex: X- axis travel 1000mm, X pitch =20mm, Testing point 50, and making Positive compensation.

If laser measuring first point 20.003, means over run 3um, then input +X 3 as first value.

If laser measuring first point 19.996, means less run 4um, then input +X -4 as first value.

6. Press "3" into the XYZ PARAMETER SYSTEM (Fig. 6)

```
   XYZ axis parameter
X axis scale (um)        :     0.625000   XYZ max velo. (mm/min)  : 99999
Y axis scale (um)        :     0.625000   XYZ min velo. (mm/min)  :    10
Z axis scale (um)        :     0.625000   Home stop dec. (M/sec^2):     2.00
+X travel (mm)           :   400.000      XYZ emerg.(dec)(M/sec^2):     0.20
-X travel (mm)           :   420.000      X home dir. (0=-,1=+)   :     1
+Y travel (mm)           :   210.000      Y home dir. (0=-,1=+)   :     0
-Y travel (mm)           :   210.000      Z home dir. (0=-,1=+)   :     1
+Z travel (mm)           :    15.000      Rapid acc(dec) (M/sec^2):     0.00
-Z travel (mm)           :   420.000      Feed  acc(dec) (M/sec^2):     0.00
X encoder (4*ppr)        :  8000
Y encoder (4*ppr)        :  8000
Z encoder (4*ppr)        :  8000
X home move (mm)         :   425.000
Y home move (mm)         :   215.000
Z home move (mm)         :    15.000
XYZ touch velo. (mm/min):  5000
XYZ esc.  velo. (mm/min):   500
XYZ home  velo. (mm/min):  5000
X index rpm.             :    10
Y index rpm.             :    10
Z index rpm.             :    10
```

FIG. 6 XYZ PARAMETER SET LIST

Description:

X asis scale(uM): X-axis pulse scale relative to space of machine . Pulley ratio 1:1 Set :0.625 Pulley ratio 1:2 Set 0.3125

Y asis scale(uM): Y-axis pulse scale relative to space of machine . Pulley ratio 1:1 Set :0.625 Pulley ratio 1:2 Set 0.3125

Z asis scale(uM): Z-axis pulse scale relative to space of machine . Pulley ratio 1:1 Set :0.625 Pulley ratio 1:2 Set 0.3125

+X travel (mm) : X-axis machine coordinate positive direction software limited length. Default value:500

-X travel (mm) : X-axis machine coordinate negative direction software limited length, Default value:500

+Y travel (mm) : Y-axis machine coordinate positive direction software limited length, Default value:250

-Y travel (mm) : Y-axis machine coordinate negative direction software limited length, Default value:250

+Z travel (mm) : Z-axis machine coordinate positive direction software limited length, Default value:15

-Z travel (mm) : Z-axis machine coordinate negative direction software limited length, Default value:500

X encoder (4*ppr): X-axis motor ppr. Default value: 8000

Y encoder (4*ppr): Y-axis motor ppr. Default value: 8000

Z encoder (4*ppr): Z-axis motor ppr. Default value: 8000

X Home move (mm): Distance from X-axis machine zero point to hardware limited Switch.

Y Home move (mm): Distance from Y-axis machine zero point to hardware limited Switch.

Z Home move (mm): Distance from Z-axis machine zero point to hardware limited Switch.

XYZ touch velo.(mm/min) : XYZ axes zero return, speed of forwarding to limited Switch, Default value:2500

XYZ esc velo.(mm/min) : XYZ axes zero return, speed of escape from limited Switch Default value: 500

XYZ home velo.(mm/min) : XYZ axes zero return, speed of escape from limited Switch to motor index searched, Default value: 5000

X index rpm: X-axis home return, speed of move to machine zero point and motor index signal searched. Default value:10

Y index rpm: Y-axis home return, speed of move to machine zero point and motor index signal searched. Default value:10

Z index rpm: Z-axis home return, speed of move to machine zero point and motor index signal searched. Default value:10

XYZ max velo. (mm/min.) : XYZ axes the highest speed , Pulley ratio→ 1:1 Default value:9000 Pulley ratio →1:2 Default Vale:4500

XYZ min velo. (mm/min.) : XYZ axes the lowest speed , Default value :10

Home stop dec. (M/sec^2): Reduction value of XYZ axes home return touch limited switch. Default value : 1

XYZ emerg. (dec) (M/sec^2): Reduction value of XYZ axes emergency stop Default value : 5

X home dir. (0=-, 1=+) : X- axis home return direction. Value 1: Positive home return

Y home dir. (0=-, 1=+) : Y- axis home return direction. Value 1: Positive home return

Z home dir. (0=-, 1=+) : Z- axis home return direction. Value 1: Positive home return

Rapid acc(dec) (M/sec^2) : Positioning (G00) acceleration. Default value: 0.7

Feed acc(dec) (M/sec^2) : Cutting Positioning acceleration. Default value: 0.3

7. Press "4" into MPG parameter (FIG. 7)

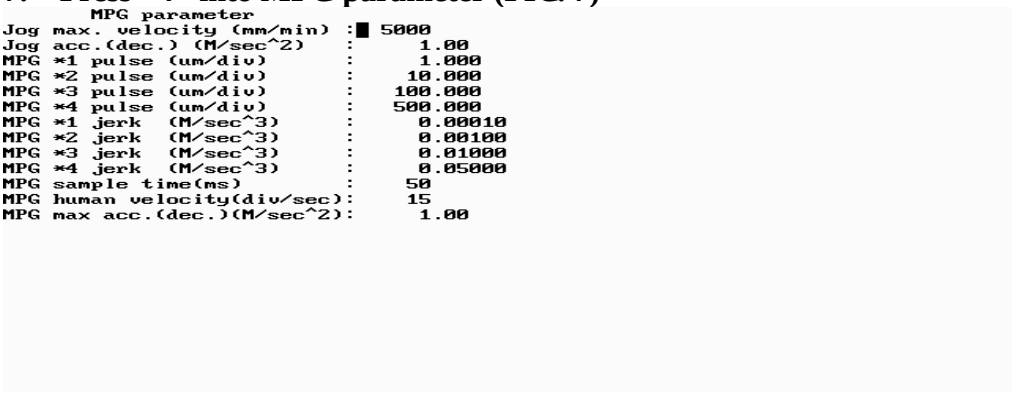

MPG parameter
Jog max. velocity (mm/min) : 5000
Jog acc.(dec.) (M/sec^2) : 1.00
MPG *1 pulse (um/div) : 1.000
MPG *2 pulse (um/div) : 10.000
MPG *3 pulse (um/div) : 100.000
MPG *4 pulse (um/div) : 500.000
MPG *1 jerk (M/sec^3) : 0.00010
MPG *2 jerk (M/sec^3) : 0.00100
MPG *3 jerk (M/sec^3) : 0.01000
MPG *4 jerk (M/sec^3) : 0.05000
MPG sample time(ms) : 50
MPG human velocity(div/sec): 15
MPG max acc.(dec.)(M/sec^2): 1.00

FIG. 7 MPG PARAMETER

Description:

Jog max. Velocity(mm/min): Remote Controller rapid moving, Default value:3000

Jog acc. (dec.) (M/sec^2) : remote controller rapid moving acceleration & reduction Default value: 1

MPG*1 pulse (um/div): Handwheel set positioning *1, per scale for machine moving distance. Default value: 1

MPG*2 pulse (um/div): Handwheel set positioning *2, per scale for machine moving distance. Default value: 10

MPG*3 pulse (um/div): Handwheel set positioning *3, per scale for machine moving distance. Default value: 50

MPG*4 pulse (um/div): Handwheel set positioning *4, per scale for machine moving distance. Default value: 100

MPG*1 jerk (M/sec^3): Handwheel per sec. scale larger than positioning parting Value, move by speed positioning type, "1" for change Acceleration, Default value: 0.0001

MPG*2 jerk (M/sec^3): Handwheel per sec. scale larger than positioning parting Value, move by speed positioning type, "2" for change Acceleration, Default value: 0.001

MPG*3 jerk (M/sec^3): Handwheel per sec. scale larger than positioning parting Value, move by speed positioning type, "3" for change Acceleration, Default value: 0.005

MPG*4 jerk (M/sec^3): Handwheel per sec. scale larger than positioning parting Value, move by speed positioning type, " 4" for change Acceleration, Default value: 0.01

MPG sample time (ms) : Handweel sample time, not necessary change. Default value :50

MPG human velocity(dis/sec): Set handwheel position type. Locating or velocity Positioning parting value. Default vale: 15

MPG max acc. (dec) (M/sec^2): Set handwheel moving the rapidest speed Default vale: 1

8. Press"5" into S axis (FIG.8)

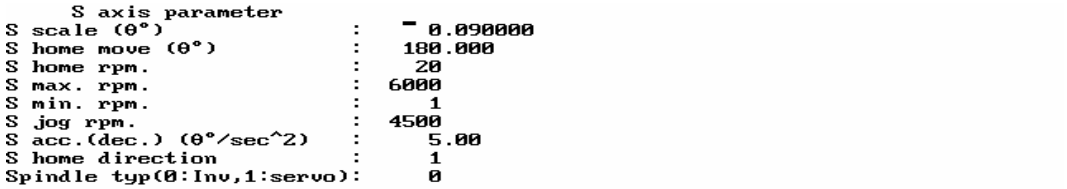

S axis parameter
S scale (θ°) : - 0.090000
S home move (θ°) : 180.000
S home rpm. : 20
S max. rpm. : 6000
S min. rpm. : 1
S jog rpm. : 4500
S acc.(dec.) (θ°/sec^2) : 5.00
S home direction : 1
Spindle typ(0:Inv,1:servo): 0

FIG. 8 S axis parameter

Description:

S scale (θ): Spindle Motor pulse scale relative to spindle rotating angle.

S home move (θ): Degree from Spindle machine zero point to hardware limited . Default Value: 180

S home rpm. : Spindle home return speed Default Value: 50

S max. rpm: Spindle max rpm. Default value: 6000

S min. rpm : Spindle min. rpm. Default value: 1

S jog rpm: Spindle for manual move max. speed. Default value: 3000

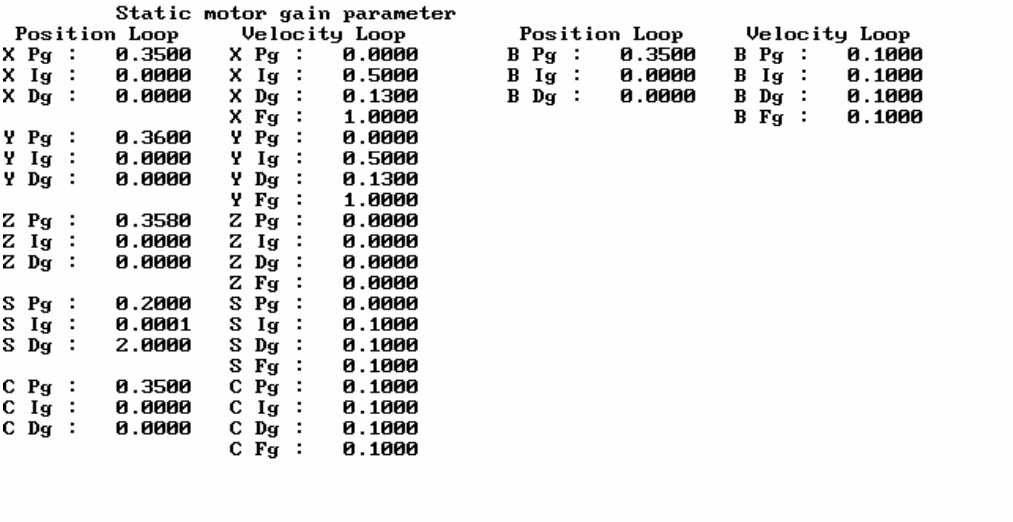
S acc. (dec.)(θ/Sec^2): Spindle rotating acceleration. Default value:20

S home direction: Spindle home return direction. Default value: 1

Spindle type(0:Inv, 1: servo): Spindle motor type. 0→Inverter

1→Servo

9. Press "6" into Static motor gain parameter(FIG. 9)



Static motor gain parameter

Position Loop		Velocity Loop		Position Loop		Velocity Loop	
X Pg :	0.3500	X Pg :	0.0000	B Pg :	0.3500	B Pg :	0.1000
X Ig :	0.0000	X Ig :	0.5000	B Ig :	0.0000	B Ig :	0.1000
X Dg :	0.0000	X Dg :	0.1300	B Dg :	0.0000	B Dg :	0.1000
		X Fg :	1.0000			B Fg :	0.1000
Y Pg :	0.3600	Y Pg :	0.0000				
Y Ig :	0.0000	Y Ig :	0.5000				
Y Dg :	0.0000	Y Dg :	0.1300				
		Y Fg :	1.0000				
Z Pg :	0.3580	Z Pg :	0.0000				
Z Ig :	0.0000	Z Ig :	0.0000				
Z Dg :	0.0000	Z Dg :	0.0000				
		Z Fg :	0.0000				
S Pg :	0.2000	S Pg :	0.0000				
S Ig :	0.0001	S Ig :	0.1000				
S Dg :	2.0000	S Dg :	0.1000				
		S Fg :	0.1000				
C Pg :	0.3500	C Pg :	0.1000				
C Ig :	0.0000	C Ig :	0.1000				
C Dg :	0.0000	C Dg :	0.1000				
		C Fg :	0.1000				

FIG. 9 Static motor gain

Description:

Separate of Position Loop (left side) & Velocity Loop (right side)

Basic on the position Loop , and the velocity Loop for up -grade space.

3 AXES position loop Default value:

XYZ axes gain : P set 0.35

I set 0

D set 0

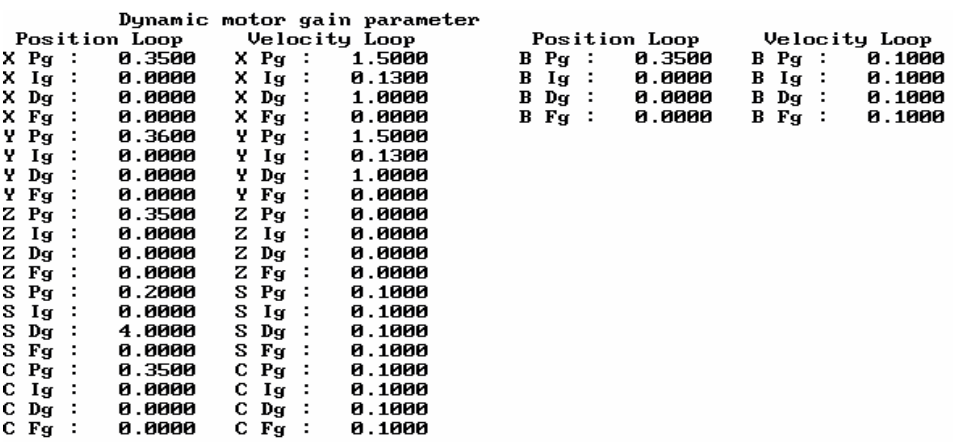
Spindle Position Loop Default value:

P set 0.2

I set 0.0001

D set 2

10. Press"7" into Dynamic motor gain (FIG. 10)



Dynamic motor gain parameter

Position Loop		Velocity Loop		Position Loop		Velocity Loop	
X Pg :	0.3500	X Pg :	1.5000	B Pg :	0.3500	B Pg :	0.1000
X Ig :	0.0000	X Ig :	0.1300	B Ig :	0.0000	B Ig :	0.1000
X Dg :	0.0000	X Dg :	1.0000	B Dg :	0.0000	B Dg :	0.1000
X Fg :	0.0000	X Fg :	0.0000	B Fg :	0.0000	B Fg :	0.1000
Y Pg :	0.3600	Y Pg :	1.5000				
Y Ig :	0.0000	Y Ig :	0.1300				
Y Dg :	0.0000	Y Dg :	1.0000				
Y Fg :	0.0000	Y Fg :	0.0000				
Z Pg :	0.3500	Z Pg :	0.0000				
Z Ig :	0.0000	Z Ig :	0.0000				
Z Dg :	0.0000	Z Dg :	0.0000				
Z Fg :	0.0000	Z Fg :	0.0000				
S Pg :	0.2000	S Pg :	0.1000				
S Ig :	0.0000	S Ig :	0.1000				
S Dg :	4.0000	S Dg :	0.1000				
S Fg :	0.0000	S Fg :	0.1000				
C Pg :	0.3500	C Pg :	0.1000				
C Ig :	0.0000	C Ig :	0.1000				
C Dg :	0.0000	C Dg :	0.1000				
C Fg :	0.0000	C Fg :	0.1000				

FIG. 10 Dynamic Motor Gain

Description:

Position Loop parameter Default value:

XYZ axes gain: P set 0.35

I set 0

D set 0

F set 0

Spindle Position Loop parameter Default value:

P set 0.2

I set 0

D set 4

F set 0

*****Velocity Loop parameter in reserve**

11. Press"8" into C axis parameter (FIG. 11)

```
          C axis parameter
C scale (θ°)                :     0.000000
C+ travel (θ°)              :   999.000
C- travel (θ°)              :   999.000
C encoder (*4ppr)           :  8000
C home speed (θ°/min)       :  1000
C home move (θ°)            :    10.000
C index rpm.                :    10
C acc.(dec.)(θ°/sec^2)      :     0.20
C max. speed (θ°/min)       :  1800
C min. speed (θ°/min)       :   100
C emg.acc.(dec.)(θ°/sec^2):     5.00
C home direction (0=-,1=+):     1
C backlash (.001 θ°)        :     7
C tolerance (.001 θ°)       :     0
C home limit (0=NO,1=NC)    :     1
```

FIG. 11 C axis parameter

Description:

C scale (θ) : C axis motor pulse scale relative to spindle rotating degree.

C+ travel(θ): C-axis machine coordinate positive direction software limited Angle. Default value: 999.000

C- travel (θ) : C-axis machine coordinate negative direction software limited angle Default value:999.000

C encoder(4*ppr): C-axis motor ppr. Default value: 8000

C home speed (θ/min): C axis zero return, speed of forwarding to limited Switch, Default value:1000

C home move (θ): Degree from C-axis machine zero point to hardware limited . Default value:10

C index rpm: C-axis home return, speed of move to machine zero point and motor Index signal searched. Default value:10

C acc(dec) (θ/sec^2) : C-axis acceleration value. Default value: 0.2

C max. speed(θ/min): C max rpm. Default value: 1800

C min. speed(θ/min): C axis min. rpm. Default value: 100

C emg.acc(dec) (θ/sec^2): Reduction value of Caxis emergent stop
Default value : 5

C home direction:C axis home return direction. Default Value :1

C backlash (.001θ):C axis backlash. Default Value :0

C tolerance (.001θ):C axis home return direction. Default Value :0

C home limit (0=NO,1=NC):C axis home limit type. Default Value :1

12. Press"9" into B axis parameter (FIG. 12)

```
          B axis parameter
B scale (0°)                :      0.000000
B+ travel (0°)              :    115.000
B- travel (0°)              :    115.000
B encoder (*4ppr)           :   8000
B home speed (0°/min)       :   1000
B home move (0°)            :    115.000
B index rpm.                :     10
B acc.(dec.)(0°/sec^2)      :      0.20
B max. speed (0°/min)       :   1800
B min. speed (0°/min)       :    100
B emg.acc.(dec.)(0°/sec^2):      5.00
B home direction (0=-,1=+):      1
B backlash (.001 0°)        :      0
B tolerance (.001 0°)       :      0
B home limit (0=NO,1=NC)    :      1
```

FIG. 12 B axis parameter

Description:

B scale (θ) : B axis motor pulse scale relative to spindle rotating degree.

B+ travel(θ): B axis machine coordinate positive direction software limited
Angle. Default value: 115.000

B- travel (θ) : B axis machine coordinate negative direction software limited angle
Default value:115.000

B encoder(4*ppr): B axis motor ppr. Default value: 8000

B home speed (θ/min): B axis zero return, speed of forwarding to limited
Switch, Default value:1000

B home move (θ): Degree from B axis machine zero point to hardware limited .
Default value:115

B index rpm: B axis home return, speed of move to machine zero point and motor
Index signal searched. Default value:10

B acc(dec) (θ/sec^2) : B axis acceleration value. Default value: 0.2

B max. speed(θ/min): B axis max rpm. Default value: 1800

B min. speed(θ/min): B axis min. rpm. Default value: 100

B emg.acc(dec) (θ/sec^2): Reduction value of B axis emergent stop
Default value : 5

B home direction:B axis home return direction. Default Value :1

B backlash (.001θ):B axis backlash. Default Value :0

B tolerance (.001θ):B axis home return direction. Default Value :0

B home limit (0=NO,1=NC):B axis home limit type. Default Value :1

13. Press"10" into Miscellaneous Parameter (FIG.13)

```
                    Miscellaneous parameter
Accleration curve          :    1          Atc type(0=Off,1...)     :     1
  (1=linear ramp,2=linear fixed time)        Tool Number              :    16
  (3=SS scurve fixed time)                   Change Tool Pos. (mm):  -154.960
Linear fixed time(ms)      :  250          Demo (0=Off,1=On)        :     1
SS curve fixed time (ms)   :  200
Costh (0-180 θ°)           :   30
Tolerance TH (0-180 θ°)    :    5
Disp.Coordi.(0:Act,1:Com):    1
Drilling clearance (mm)    :    2.000
Max. d-well time (sec)     :   10.000
Velocity refresh time(ms):   50
Tolerance X (um)           :   15
Tolerance Y (um)           :   15
Tolerance Z (um)           :    0
Tap acc.(dec.)(θ°/sec^2)   :   50.00
Tap emerg(dec)(θ°/sec^2)   :   20.00
Tap MPG pulse (um/div)     :   10.000
Tap MPG jerk  (M/sec^3)    :    0.00100
Servo loop (0:P 1:P+V)     :    0
Tap  (0=Off,1=On)          :    1
Bore (0=Off,1=On)          :    1
```

FIG. 13 Miscellaneous Parameter

Description:

Acceleration Curve:

(1. Linear acceleration curve, 2. Fixed time linear acceleration curve,

3. Fixed time SS acceleration curve , use 1, Default value :1)

Linear fixed time (ms): Default value :250

SS curve fixed time (ms): Default Value : 200

Costh (0-180 Drg.) : machining present linear and next linear, projecting to X-axis speed larger than set value, X-axis acceleration/reduction for machining next linear. Default value : 30

Tolerance TH (0-180Drg.): Default Value: 5

Disp. Coordi.(0:Act, 1:Com): Coordinate "0" display the Present value, "1" display the common value.

Drilling clearance (mm): G73 for backward, G83 for reduction. Default value : 2

Max. d-well time (sec) : G04 suspend program, max time Default value: 10

Velocity refresh time (ms): Machining speed screen refresh time. Default value: 50

Tolerance X (um): 3D CAM Machining, X-axis movement less than set vale, X-axis not reacting. Default value: 15

Tolerance Y (um): 3D CAM Machining, Y-axis movement less than set vale, Y-axis not reacting. Default value: 15

Tolerance Z (um): 3D CAM Machining, Z-axis movement less than set vale, Z-axis not reacting. Default value: 0

Tap.acc.(dec.)(θ/Sec^2): Acceleration / Reduction of Tapping. Default value: 30

Tap emerg.(dec.)(θ/sec^2):Acceleration / reduction for Tapping emergent stop. Default Value: 20

Tap MPG pulse (um/div): For Tapping ,Per scale of Handwheel cause machine moving distance. Default value: 10

Tap MPG jerk (M/sec^3): For tapping, Handwheel per scale of positioning change Acceleration value. Default value: 0.001

Servo loop (0:P, 1: P+V) : 0 as positioning loop, 1 as velocity loop Default value: 0

Tap (0=off, 1=on) : With/Without Tapping function. 1 means with tapping Function.

Bore(0=off, 1=on): With/without Boring function. 1 means with Boring Function.

ATC type (0=off, 1=…) With ATC system. "0" Without ATC

Tool number: With ATC system for set Tool number.

Change tool pos. (mm): With ATC system, Z-axis tool changes position.

Demo (0=off, 1=on) : 1 use on PC DEMO, 0 for machine use.

14. Press "11" into DNC parameter (FIG. 14)

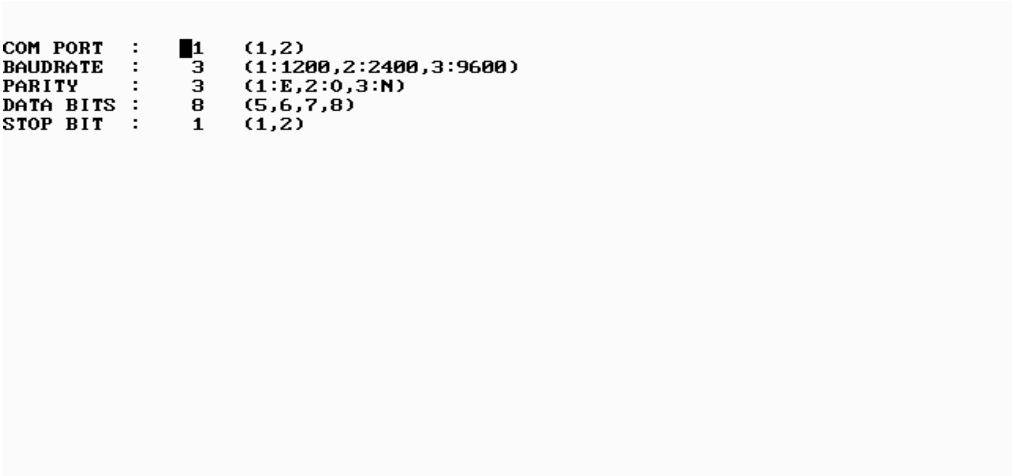


FIG. 14 DNC PARAMETER

Description:

COM PORT : Set for DNC link . Default Value: 1

BAUDRATE : Set DNC Link transfer rate. Default value : 3

PARITY: Set DNC Link parity . Default value :3

DATA BITS: Set DNC Link Data Bits Default value: 8

STOP BIT: Set DNC Link Stop bit. Default value: 1

15.Press "12" into I/O TEST (FIG. 15)

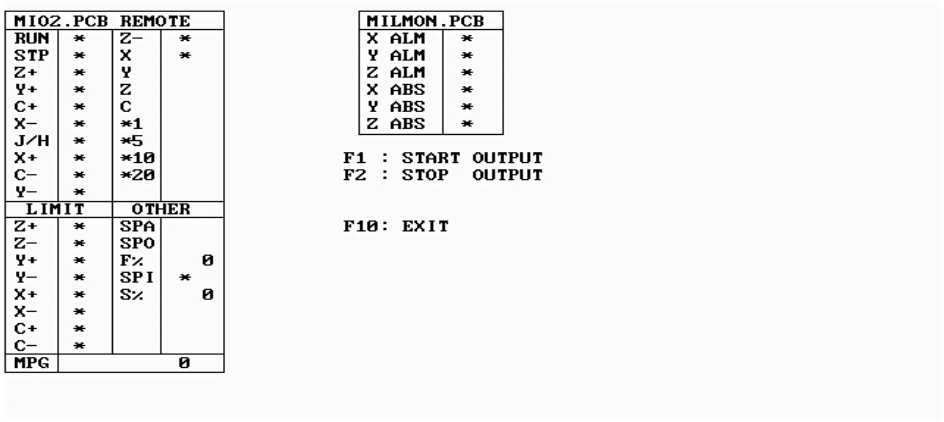


FIG. 15 I/O TEST

Description:

I/O Test for testing I/O switch, signal read by PC, including remote box switch Switch on each axis, rate adjust button ,Spindle speed button. (According to Machine Model)

16. Press "13" into motor tune function (FIG. 16)

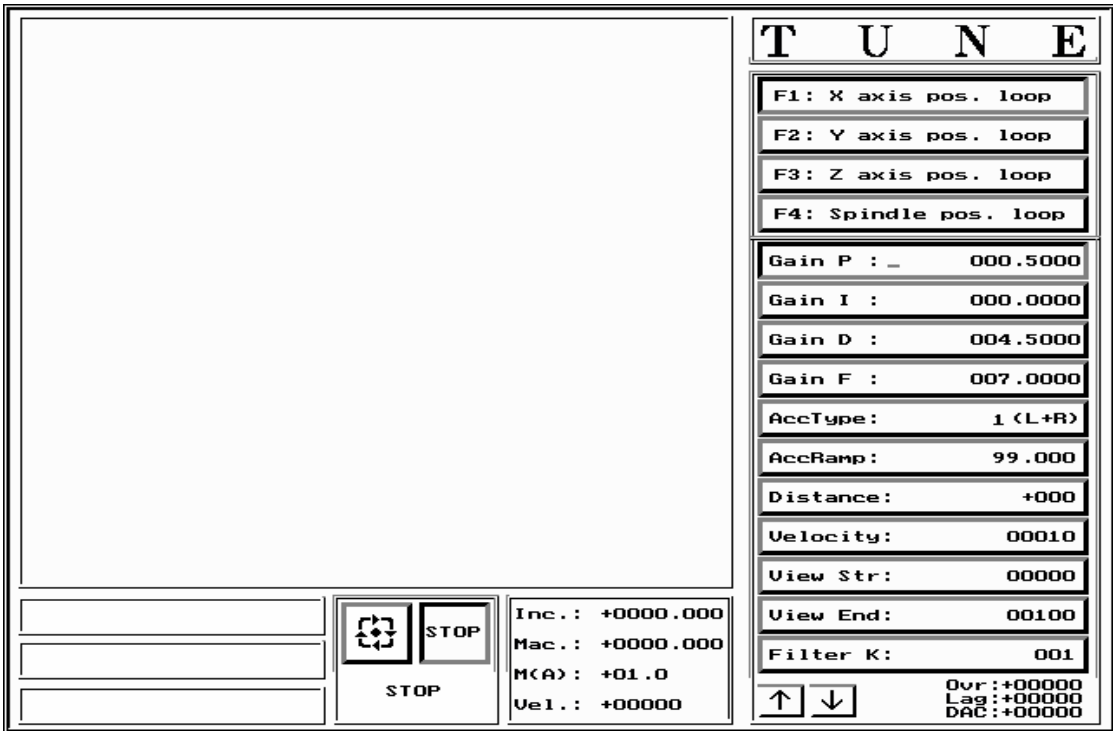


FIG. 16 Motor Parameter

Description

For adjusting and testing motor parameter gain number. After testing set the number to Static motor Gain and Dynamic motor Gain.

17. Press "14" into ATC parameter (FIG. 17)

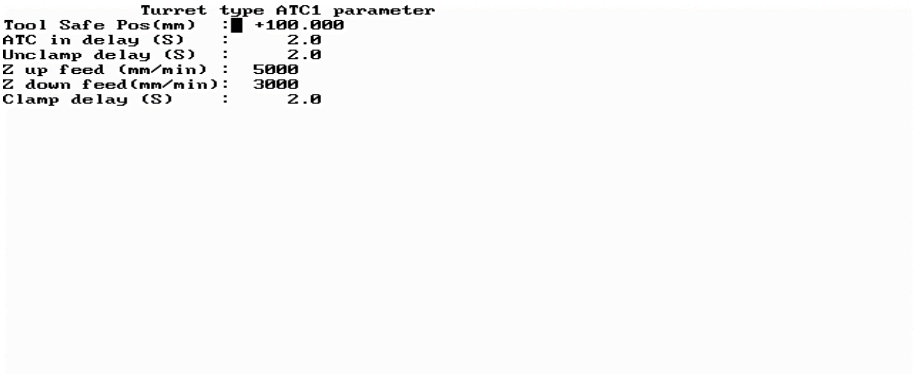

Turret type ATC1 parameter
Tool Safe Pos(mm) : +100.000
ATC in delay (S) : 2.0
Unclamp delay (S) : 2.0
Z up feed (mm/min) : 5000
Z down feed(mm/min): 3000
Clamp delay (S) : 2.0

FIG. 17 ATC Parameter

Description:

Tool Safe pos(uM) : Move Z axis to this position before magazine rotate.

ATC in delay (S) : Delay for magazine in. Default value : 2

Unclamp delay (S) : Delay for unclamp. Default value :2

Z up feed (mm/min) : Z axis move up speed. Default value: 5000

Z down feed (mm/min) : Z axis move down speed. Default value: 3000

Clamp delay (S) : Delay for clamp. Default value :2

18. Press "15" into MOTOR parameter (FIG. 18)

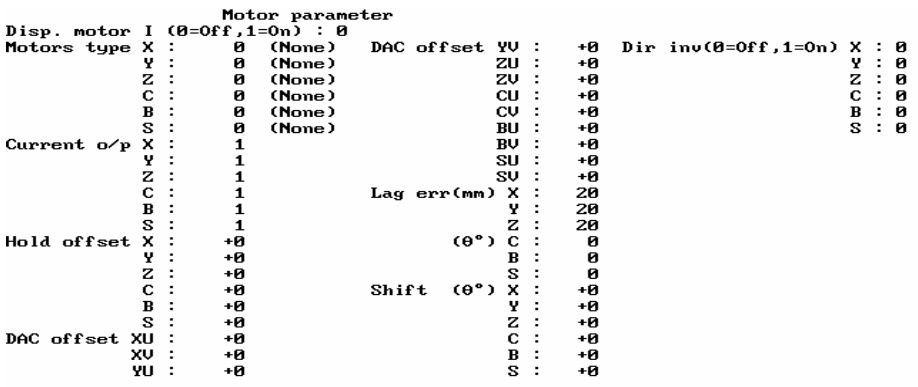


FIG. 18 MOTOR Parameter

Description:

Motors type : Define motor type for each axis.

Current o/p : Define output current for each axis.

Hold offset : Motor balance adjust.

DAC offset : Command output voltage offset.

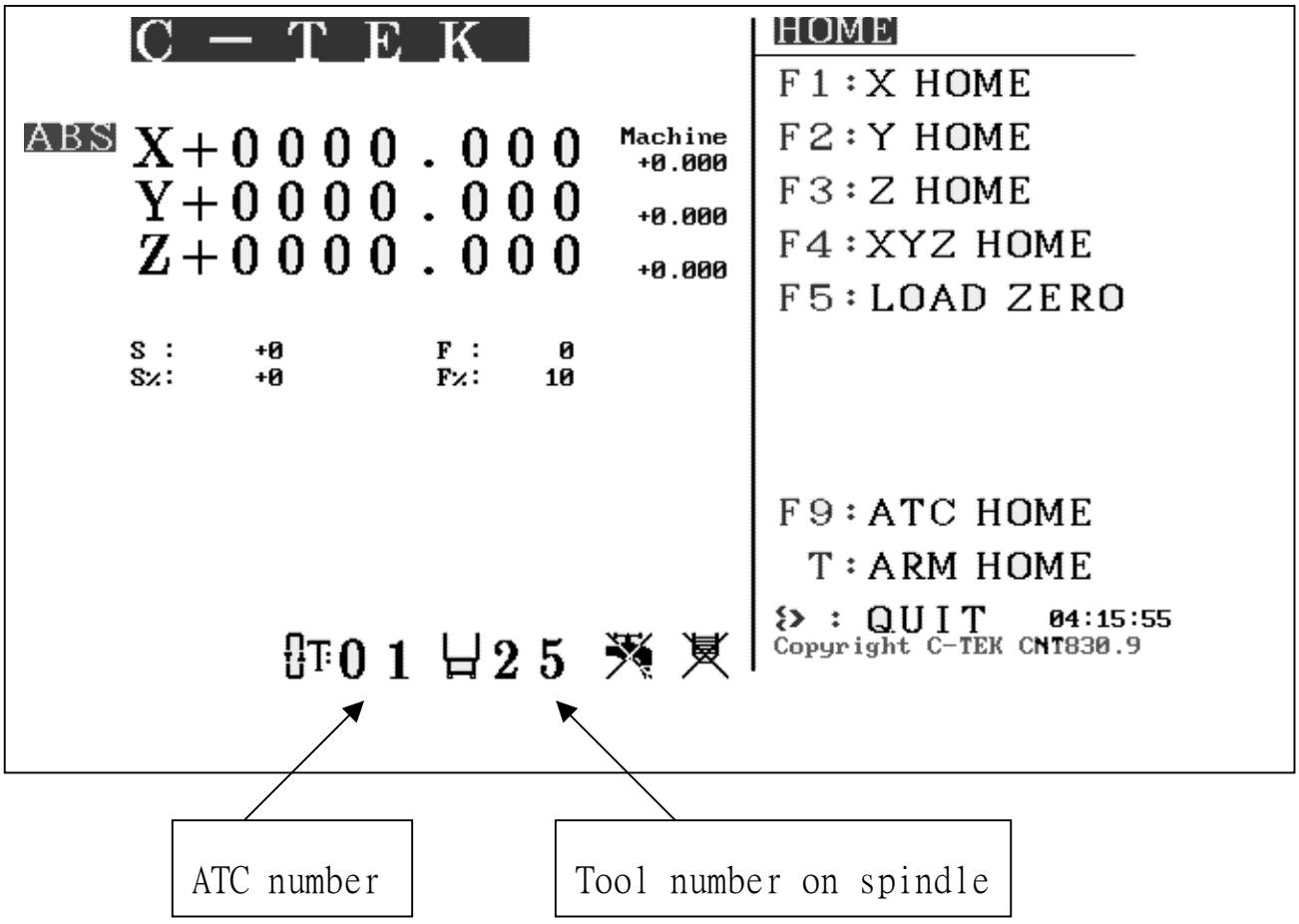
Lag err : Command position and encoder feedback position error value. If actual value bigger than setting value. It will show “lag error” on the screen.

Shift : For motor UVW phase offset.

Dir : Motor rotate direction.

APPENDIX 二. ARM ATC OPERATION MANUAL

HOME ：



1. While tool change's error cause re-start machine. ATC number and screen show different number. Press ATC HOME to correct it.

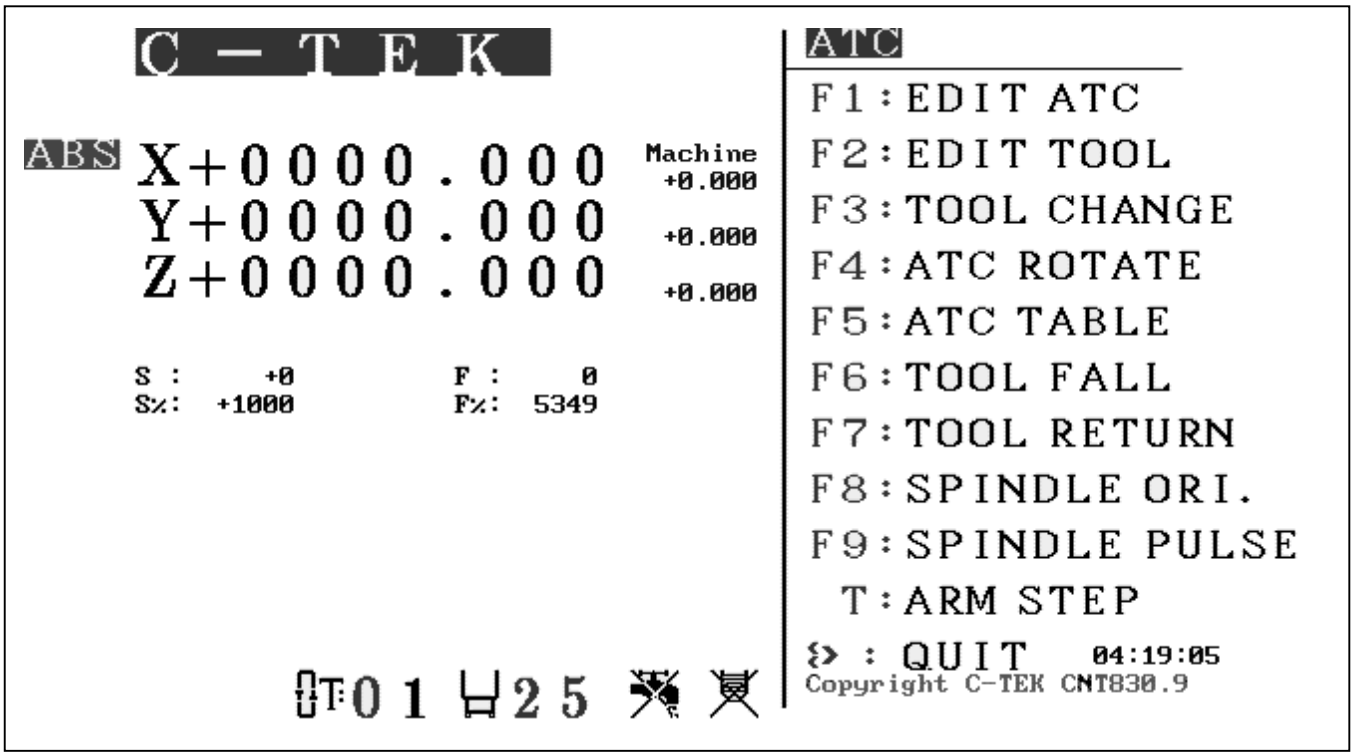
2. While tool change's error cause arm stop in incorrect position. The screen will show "--ARM wrong position Alarm—". Press T:ARM HOME to correct it.

ACTION：

F9:ATC HOME : Press F9 to run ATC HOME. ATC will back to NO.1.
Please note when execute ATC HOME. The pot must be in Tool-returning.

T : ARM HOME : Press T to run ARM HOME. The ARM will back to home position.
Please move Z axis to top and take tool out from spindle.

F2:MANUAL->F9:ATC



Detail：

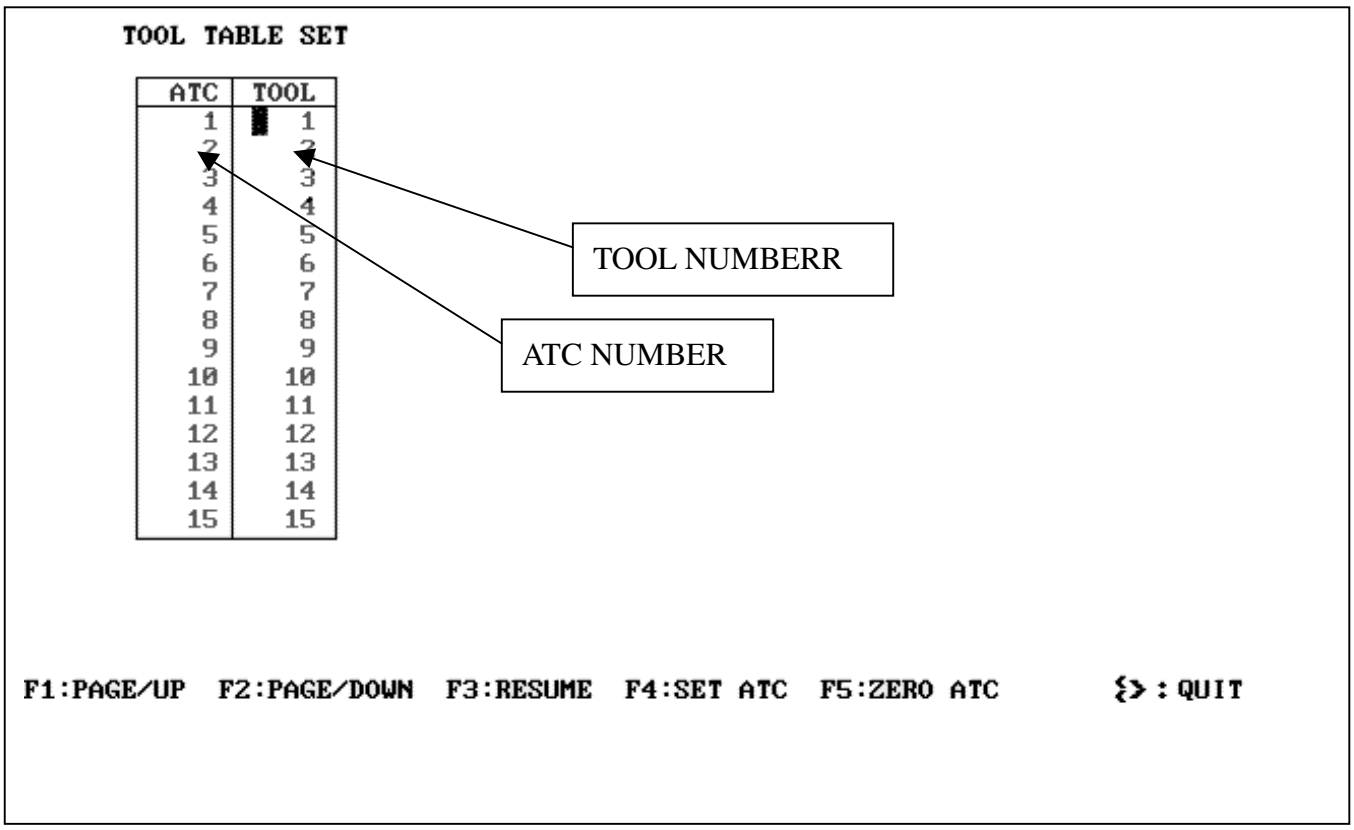
F4:ATC ROTATE : When press F4 without press **"-"** in two seconds. The ATC turn clockwise and add one tool. Press **"-"** within two seconds. The tool turn counterclockwise and subtract one tool.

F6:TOOL FALL : Press F6. The ATC plate will be fall.
This function can use to install tools by hand.
Please note that before quit MANUAL function, the machine must be excute F7:Tool-returning.

F7:TOOL RETURN : Press F7. The ATC plate will back to original position.

T :ARM STEP : Press T to execute ARM step. The step time according to system parameter **No"14.TURRET1 Parameter"-> "ATC ARM ROTATE (sec.):0.5"**. The default is 0.5 second.

F5:ATC TABLE : Press F5 show this table

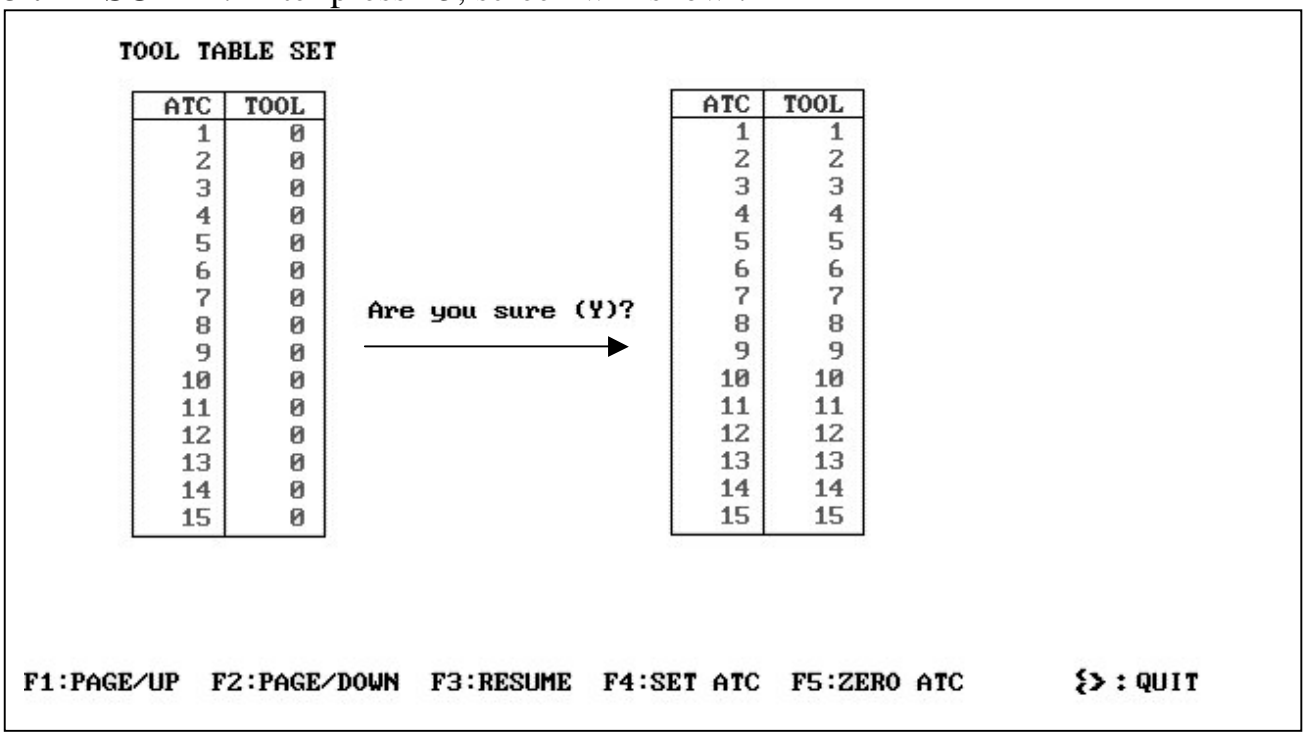


Left is ATC number. Right is Tool number. You can move cursor to edit.
Please note:

1. Tool number can not repeat.
2. If ATC number be set in fixed (big tool). The before and after this ATC tool number must be set to 0. And the before and after this ATC number can not be set in fixed ATC.
3. When ATC show tool number is 0. It means that no tool in the ATC number.
4. If ATC set several fixed tools(big tool) and several small tools. The program will recognize it and automatic to change them. Big diameter tool only can be changed with big diameter in ATC.
   And small diameter tool only can be changed with small diameter too in ATC.
5. Please note if spindle using big tool. The ATC have to be set one fixed tool and it's tool number must be 0.

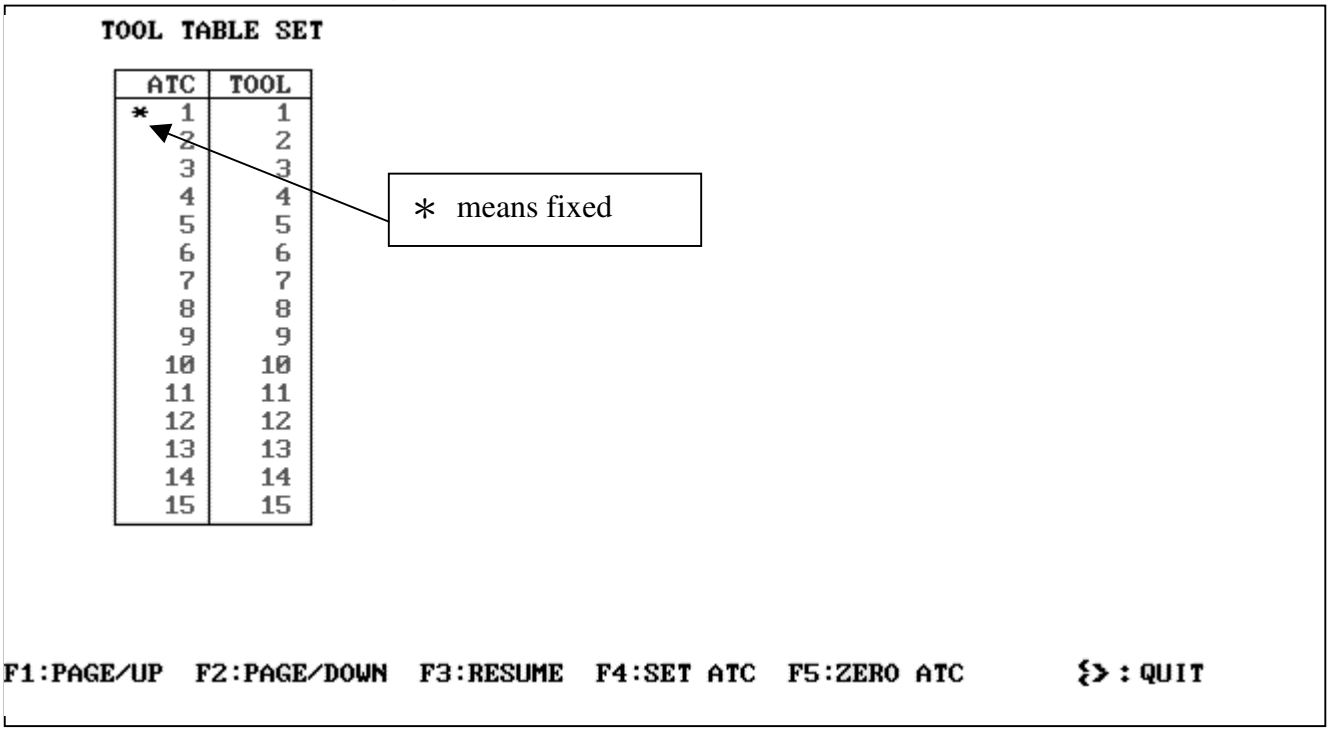
Other Function in ATC Table edit :

F3 : RESUME : After press F3, screen will show :



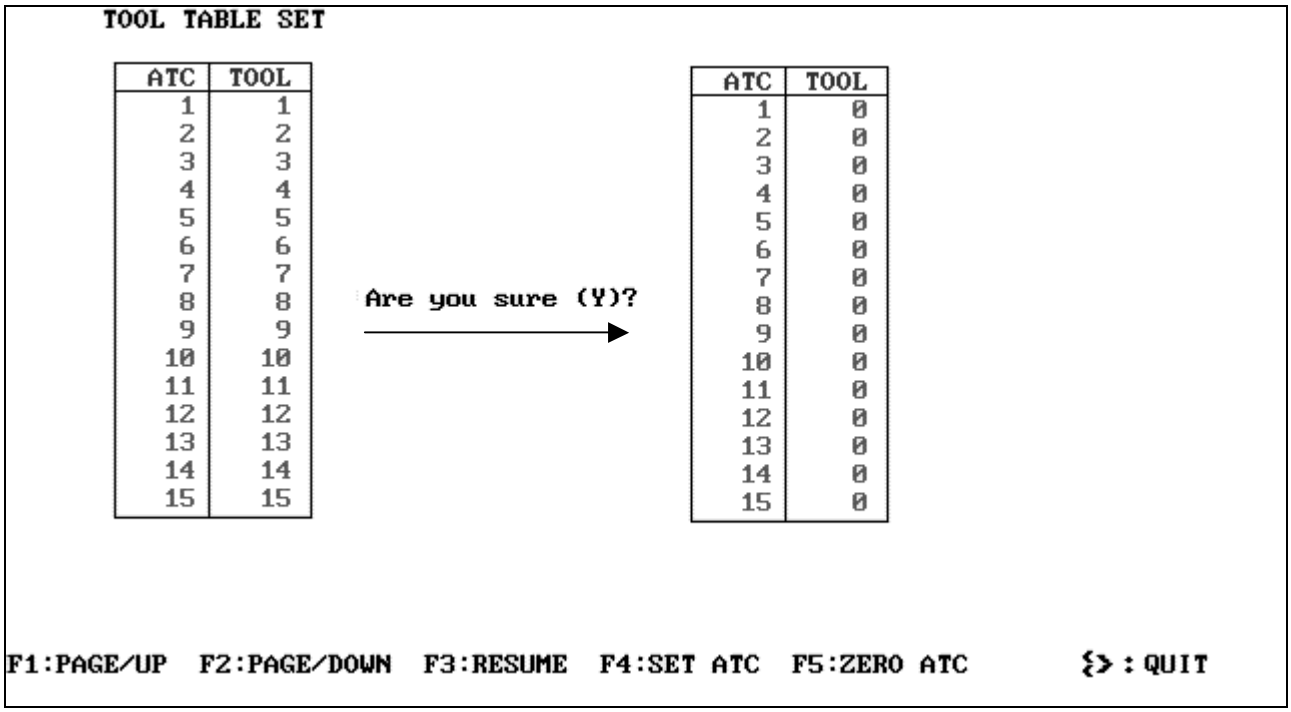
Above mention. Left is original list. PressF3, screen will show”Are you sure(Y)?”. Press”Y”, all tool number will be orderd from 1 to 15 like right list. Press”N” and nothing will be changed.

F4 : SET ATC : Press F4, screen will show :



∗ Please note : If this tool set in fixed tool (big tool) the next tool (before and after) number must be set 0. For example : set ATC no.1 in fixed tool, and ATC no.2 and no.24 must be 0.

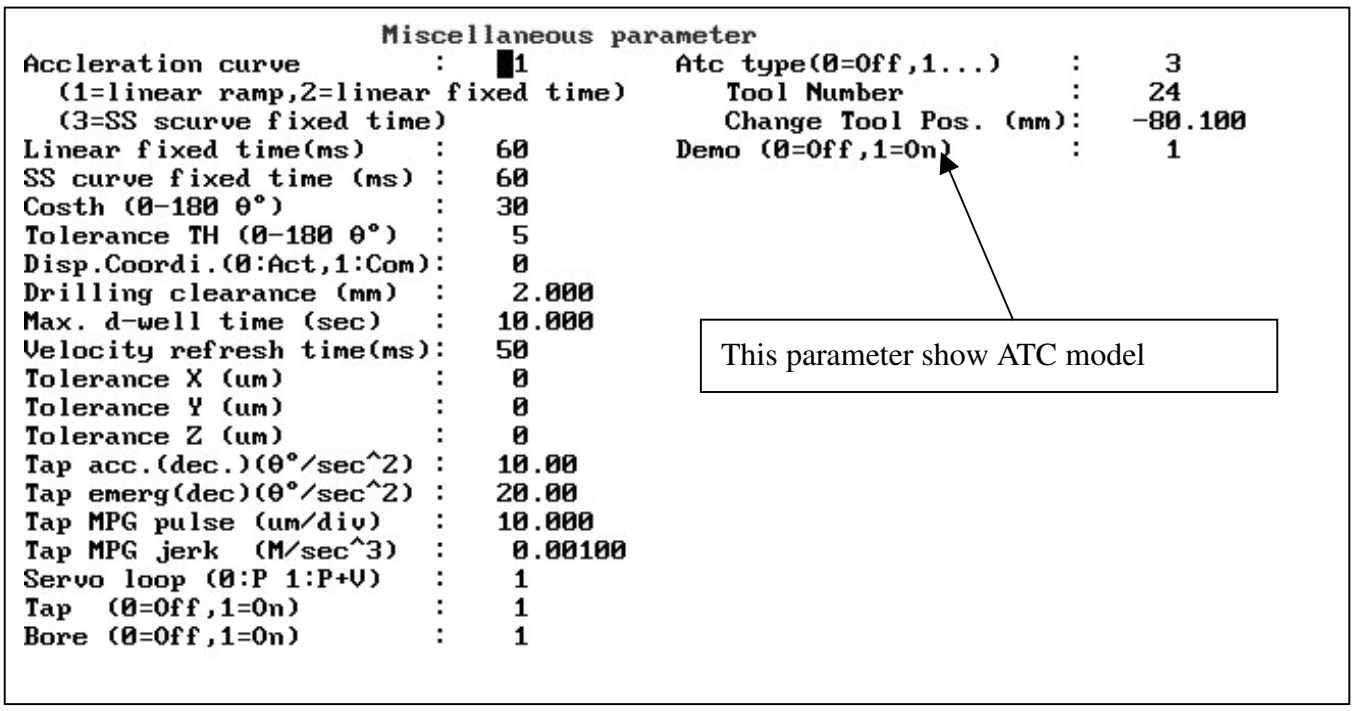
F5: ZERO ATC : Press F5



In above. Left is original list. Press F5, the screen will show ”Are you sure” and press ”Y”. All tool’s number will become ”0 ”. Press ”N ” and nothing will be changed.

New function in system parameter:

10.Miscellaneous Parameter


Miscellaneous parameter
Accleration curve : 1
(1=linear ramp,2=linear fixed time)
(3=SS scurve fixed time)
Linear fixed time(ms) : 60
SS curve fixed time (ms) : 60
Costh (0-180 0°) : 30
Tolerance TH (0-180 0°) : 5
Disp.Coordi.(0:Act,1:Com): 0
Drilling clearance (mm) : 2.000
Max. d-well time (sec) : 10.000
Velocity refresh time(ms): 50
Tolerance X (um) : 0
Tolerance Y (um) : 0
Tolerance Z (um) : 0
Tap acc.(dec.)(0°/sec^2) : 10.00
Tap emerg(dec)(0°/sec^2) : 20.00
Tap MPG pulse (um/div) : 10.000
Tap MPG jerk (M/sec^3) : 0.00100
Servo loop (0:P 1:P+V) : 1
Tap (0=Off,1=On) : 1
Bore (0=Off,1=On) : 1
Atc type(0=Off,1...) : 3
Tool Number : 24
Change Tool Pos. (mm): -80.100
Demo (0=Off,1=On) : 1
This parameter show ATC model

ATC TYPE :

0 : NO ATC

1 : ARMLESS TYPE ATC

2 : DOUBLE COLUMN CNC MILLING MACHINE ATC

3 : ARM TYPE ATC

14.TURRET1 Parameter

```
            Turret type ATC1 parameter
Tool Safe Pos(mm)  :     +5.000
ATC in delay (S)   :      1.0
Unclamp delay (S)  :      1.0
Z up feed (mm/min) :  5000
Z down feed(mm/min):  3000
Clamp delay (S)    :      1.0
Y Safe Pos(mm)     :  +900.000
  Tool Pos.(mm)    :  +950.000
  Feed (mm/min)    :  3000
ATC door sw. (0=N,1=Y) : 1
    tool det.(0=N,1=Y) : 1
ATC-ARM rotate (sec.) :  1.0
    ARM failed (sec.) :  3
    ARM check 0 tool  :  1
```

Y Safe Pos(mm) :Tool change safe position for Y axis. It's for CNC twin column machine.

Tool Pos.(mm) :Tool change position for Y axis. It's for CNC twin column machine.

Feed (mm/min) :Tool change speed for Y axis. It's for CNC twin column machine.

ATC door sw. (0=N,1=Y) : Set CNC twin column machine ATC install splash and door switch, 0=NO,1=YES.

tool det.(0=N,1=Y) : Does ATC install empty tool detect, 0=NO,1=YES.

ATC ARM ROTATE (sec.) : ATC ARM rotate time while in F2:Manual mode. Default is 0.1

ARM failed (sec.) : If ATC does not change tool in setting time in every action. Alarm will show on screen. Default is 3secs.

ARM check 0 tool : In tool table edit function. When big diameter tools set in ATC. The before and after tool number must be 0. This parameter control how many tools will be set to zero. Default is 1.